University of New Hampshire InterOperability Laboratory
Gigabit Ethernet Consortium

# Superseded

As of July 14th, 1998 the Gigabit Ethernet Consortium Clause 37 Auto Negotiation Conformance Test Suite Version 1.1 has been superseded by the release of the Clause 37 Auto Negotiation Conformance Test Suite Version 1.3. This document along with earlier versions, are available on the Ethernet Consortium test suite archive page.

Please refer to the following site for both current and superseded test suites:
http://www.iol.unh.edu/testsuites/ge/

# GIGABIT ETHERNET

**Clause 37 Auto-Negotiation**
**Test Suite**
**Version 1.1**

*Technical Document*

*Last Updated: April, 1998*


***InterOperability Laboratory***
***Research Computing Center***
***University of New Hampshire***

***121 Technology Drive, Suite 2***
***Durham, NH 03824***
***Phone: (603) 862-4822***
***Fax: (603) 862-0898***

http://www.iol.unh.edu/consortiums/index.html

### #37.1.1 Configuration Ordered Set Format

**Purpose:** To verify that the Config_Reg is properly encoded into the Configuration Ordered Sets and that the ordered sets are properly transmitted.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - Figure 36-3 PCS reference diagram, Table 36-3 Defined ordered_sets, Figure 36-6 PCS transmit code-group state diagram, Figure 36-7a PCS receive state diagram, part a, and subclause 37.2.1 Config_Reg encoding.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** Two Configuration Ordered Sets are defined, /C1/ and /C2/. /C1/ consists of /K28.5/D21.5/Dx.y /Da.b. /C2/ consists of /K28.5/D2.2/Dx.y/Da.b. Where Dx.y contains tx_Config_Reg<D7:0>, and Da.b contains tx_Config_Reg<D15:8> (refer to Table 36-3 and Figure 36-6).

The sixteen bit value of the Config_Reg base page is encoded logically as shown below:

| **LSB** | | | | | | | | | | | | | | | **MSB** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 |
| rsvd | rsvd | rsvd | rsvd | rsvd | FD | HD | PS1 | PS2 | rsvd | rsvd | rsvd | RF1 | RF2 | Ack | NP |

Reserved (rsvd) bits should be sent as 0. Of the other bits, at least full duplex (FD) or half duplex (HD) support must be indicated, by setting a value of 1. Additionally, the ACK bit should initially be 0, and then set to 1 after the DUT receives three consecutive and identical non-zero Config_Reg values (refer to test 37.3.1). As shown in Figure 36-3, the bit-order, on the medium, of Config_Reg is bits D7 to D0, then bits D15 to D8. As can be seen in Figure 36-6, the transmission of the configuration ordered sets should alternate between /C1/ and /C2/.

This test verifies that /C1/ and /C2/ are properly formed, that they are transmitted in the proper order, and that the Config_Reg is properly mapped into the Ordered Set.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by restarting auto-negotiation by any means.
2. Provide a link of /C/ codes with a Config_Reg of all zero.
3. Observe transmissions from the DUT up to and including the transmission of a non-zero Config_Reg.
4. Provide a link of /C/ codes with a non-zero Config_Reg (with ACK set to 0).

**Observable Results:**

a) Upon restart, the device should only source four byte ordered sets alternating in form between /C1/ and /C2/.
b) The non-zero Config_reg should be mapped as (MSB first) Dx.y=?HF00000 and Da.b=?A??000?, where '?' may be 0 or 1 dependent on the devices advertised abilities. 'H' and 'F' are the half/full duplex bits, one of which must be 1. 'A' is the Ack bit, which should initially be 0, and then change to 1 after step (4).

**Possible Problems:** None.

## #37.1.2 Ill-formed and Invalid /C/ Code Handling

**Purpose:** To verify that invalid codes, received while xmit=(CONFIGURATION+IDLE), cause auto-negotiation to restart, and that only Ordered Sets that are four bytes in length, that begin with /K28.5/D21.5+D2.2/ are treated as valid configuration ordered sets.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - Figure 36-3 PCS reference diagram, Table 36-1 Valid data code-groups, Table 36-2 Valid special code-groups, Figure 36-6 PCS transmit code-group state diagram and Figure 36-7a PCS receive state diagram, part a.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** April 17, 1998

**Discussion:** The proper reception and decoding of configuration ( /C/ ) ordered sets is accomplished via the PCS receive state diagram, part a (Figure 36-7a). The 16 bit Config_Reg, mapped into the four byte ordered sets at the transmitter, will only be decoded properly at the receiver if one of two valid /C/ codes are received. The defined configuration ordered sets are, /C1/ and /C2/. /C1/ consists of /K28.5/D21.5/Dx.y/Da.b/ and /C2/ consists of /K28.5/D2.2/Dx.y/Da.b/. Where Dx.y contains tx_Config_Reg<D7:0>, and Da.b contains tx_Config_Reg<D15:8>. When completing the auto-negotiation function, the transmission of idle ( /I/ ) ordered sets is required, hence Figure 36-7a permits the reception of /I/ codes. Specifically, a code pattern of /K28.5/D*!(D21.5+D2.2)/K28.5 etc. is acceptable. Any other received patterns while xmit≠DATA should result in an RX_UNITDATA.indicate(INVALID) message, which should cause auto-negotiation to restart at the receiving station. This process helps to ensure that a link is not established if the channel has a high bit error rate.
This test is designed to verify that the DUT only decodes the 16 bit Config_Reg from valid configuration ordered sets.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**
1. Initialize by restarting Auto-Negotiation by any means.
2. Continuously provide one of the stream of /C/ codes noted below.
3. Observe the Config_Reg transmitted by the DUT, monitoring bit 14 (ACK).
4. Repeat steps (1) through (3) until all patterns noted below have been tested.

Note - Test patterns may include (but need not be limited to):
Part A:
1. /K28.5/D21.5/D0.3/D0.0 …
2. /K28.5/D2.2/D0.3/D0.0/K28.5/D2.2/D0.3/D0.0 …

Part B:
1. /K28.5/D21.5/K28.5/D21.5 …
2. /K28.5/D2.2/K28.5/D2.2 …
3. /K28.5/D21.5/D0.3/K28.5/D2.2/D0.3/K28.5/D21.5/D0.3/K28.5/D2.2/D0.3 …
4. /K28.5/D21.5/D0.3/D0.0/D0.3/K28.5/D2.2/D0.3/D0.0/D0.3/K28.5/D21.5/D0.3/D0.0/D0.3/K28.5/D2.2/D0.3/D0.0/D0.3 …
5. /K28.5/D0.3/D0.0 …
6. /K28.5/D16.2/K27.7/D16.2/K28.5/D16.2/K27.7/D16.2 …
7. /K27.7/D16.2/K27.7/D16.2 …
8. /K27.7/D21.5/D0.3/D0.0/K27.7/D2.2/D0.3/D0.0/K27.7/D21.5/D0.3/D0.0/K27.7/D2.2/D0.3/D0.0 …
9. (Ending RD+) /+K28.5/+D2.2/-D0.3/+D0.0/+K28.5/-D21.5/-D0.3/+D0.0 … {D2.2 RD is flipped}
10. (Ending RD+) /+K28.5/-D21.5/+D0.3/-D0.0/-K28.5/+D2.2/-D0.3/+D0.0 … {1st D0.3 RD is flipped}
11. (Ending RD+) /+K28.5/-D21.5/-D0.3/-D0.0/-K28.5/+D2.2/-D0.3/+D0.0 … {1st D0.0 RD is flipped}
12. (Ending RD+) /+K28.5/-D21.5/-D0.3/+D0.0/-K28.5/+D2.2/-D0.3/+D0.0/+K28.5/-D21.5/-D0.3/+D0.0/+K28.5/-D2.2/+D0.3/-D0.0/-K28.5/+D21.5/+D0.3/-D0.0/-K28.5/+D2.2/-D0.3/+D0.0 … {1st -K28.5 is invalid, should be +K28.5}
13. (Ending RD+) /+K28.5/-D21.5/-D0.3/+D0.0/+K28.5/-D2.2/+D0.3/-D0.0/+K28.5/-D21.5/-D0.3/+D0.0/+K28.5/-D2.2/+D0.3/-D0.0/-K28.5/+D21.5/+D0.3/-D0.0/-K28.5/+D2.2/-D0.3/+D0.0 … {3rd +K28.5 is invalid, should be -K28.5}
14. (Ending RD-) /K28.5/D21.5/D0.3/K28.5/D2.2/D0.3 … {sequence begins with RD- but ends with RD+}

Note:
Ending RD+/- indicates that the current running disparity is either + or – when the pattern is constructed.
/+K28.5/ refers to the 10-bit code group for K28.5 from the Current RD+ column of Table 36-2.

**Observable Results:**
a) Test patterns (1) and (2) should result in the device setting bit 14 (ACK) to one after three repetitions of the pattern.
b) Test patterns (3) through (14) should result in the device restarting auto-negotiation, resulting in the transmission of a Config_Reg of all zero. Continuous reception of these patterns should prevent the DUT from ever transmitting a non-zero Config_Reg

**Possible Problems:** None.

**#37.2.1 Base Page Advertised Abilities**

**Purpose:** To verify that the DUT transmits acceptable technology abilities, and proper initial values for reserved, RF, ACK and NP bits.

**References:**

- IEEE 802.3z/D4.1, February 5, 1998 - subclause 37.2.1 Config_Reg encoding, subclause 37.2.2 Transmit function requirements, Table 37-4 Pause priority resolution, Table 37-5 AN advertisement register bit definitions.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** The sixteen bit value of the Config_Reg base page is encoded as shown below:

| **LSB** | | | | | | | | | | | | | | | **MSB** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 |
| rsvd | rsvd | rsvd | rsvd | rsvd | FD | HD | PS1 | PS2 | rsvd | rsvd | rsvd | RF1 | RF2 | Ack | NP |

The DUT should always transmit reserved bits (rsvd) as zero, and initially the acknowledge bit (Ack) should be transmitted as zero as well.
The DUT may transmit abilities that it possess by setting the appropriate bits to one, such as full duplex (FD), half duplex (HD), and Pause/Asymmetric Pause capability (PS1/PS2) (refer to table 37-4 for a complete breakdown of Pause mode resolutions). While the DUT might not transmit abilities it does possess, it shall not, at any time, transmit abilities it does not possess.
If the DUT has sensed a fault condition, it may indicate such an event via the remote fault bits (RF1/RF2).
If the DUT possesses next pages that it desires to send, then the next page bit (NP) should be set to one.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**
1. Initialize by restarting auto-negotiation by any means.
2. Provide a link of /C/ codes with a Config_Reg of all zero.
3. Observe transmissions from the DUT up to and including the transmission of a non-zero Config_Reg.

**Observable Results:**
a) The DUT should not advertise any abilities that it does not possess.
b) The reserved bits should be transmitted as zero.
c) The NP bit should be transmitted as one only if the DUT has next pages to transmit

**Possible Problems:** None.

### #37.2.2 Link Timer

**Purpose:** To verify that the implemented link_timer is within 10ms to 20ms.

**References:**

- IEEE 802.3z/D4.1 February 26, 1998 - subclause 37.3.1.1 Variables, subclause 37.3.1.4 Timers, Figure 37-6 Auto-Negotiation state diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** The Clause 37 Auto-Negotiation function includes only one timer, link_timer, which serves the primary purpose of allowing for sufficient delay such that higher-level management may write and read registers necessary for base page and next page exchange. Link_timer is specified in 37.3.1.4 to be 10ms, however a tolerance of up to +10ms is acceptable. As can be seen in the Auto-Negotiation state diagram, link_timer is required to complete at least three times prior to completion of the protocol. During this period, if an invalid code is received, auto-negotiation shall restart. In this way, link_timer also helps to prevent a link from being established on channels with exceptionally high bit-error rates.
This test endeavors to measure the implemented value of link_timer in all five locations that it is used. Refer to the state machine transitions in Figure 37-6 from AN_RESTART to ABILITY_DETECT, from COMPLETE_ACKNOWLEDGE to IDLE_DETECT, from IDLE_DETECT to LINK_OK, and from COMPLETE_ACKNOWLEDGE to NEXT_PAGE_WAIT. The fifth is specified in the an_sync_status variable in 37.3.1.1.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by restarting auto-negotiation by any means.
2. Queue a frame to transmit from the DUT.
3. Provide a link of /C/ codes with a valid Config_Reg with ACK=1 and NP=1.
4. If the DUT does not support/desire Next Page exchange, skip to step (7). Time from when the DUT sources a non-zero Config_Reg to when a Config_Reg is sourced with bit D11 toggled.
5. Provide a stream of /C/ codes to the DUT, with bit D11 toggled from the previous Config_Reg value and ACK=1. Once the DUT enters COMPLETE_ACKNOWLEDGE, repeat until the DUT sources a Config_Reg with NP=0.
6. Time from the last time bit D11 changed in the Config_Reg sourced by the DUT, to when the DUT sources /I/ codes. Skip to step (8).
7. For devices not desiring Next Page exchange, time from the first time bit D14 (Ack) is set to one in the Config_Reg sourced by the DUT, to when the DUT sources /I/ codes.
8. Once the DUT sources /I/ codes, provide a link of /I/ codes to the DUT.
9. Time from when the DUT sources the first /I/ code, to when the frame queued in (2) is sourced.
10. Now that the link is up, continue sending /I/ codes, introducing a limited number (x) of consecutive invalid codes periodically in the stream to the DUT.
11. Repeat step (6), increasing the (x) until auto-negotiation restarts due to an_sync_status=FALSE.
12. Time the duration the DUT sources a Config_Reg of all zero.

**Observable Results:**

a) The time measured in step (4) should establish an upper bound on the link_timer value used from the COMPLETE_ACKNOWLEDGE state to the NEXT_PAGE_WAIT state.
b) The time measured in step (6) or (7) should establish an upper bound on the link_timer value used from the COMPLETE_ACKNOWLEDGE state to the IDLE_DETECT state.
c) The time measured in step (9) should establish an upper bound on the link_timer value used from the IDLE_DETECT state to the LINK_OK state.
d) The number of invalid codes required in step (10) and (11) should establish the link_timer value used for an_sync_status, taking into account the requirements of Fig 36-9, Synchronization State Diagram. Note that link_timer = (x) + 16ns since 4 invalid code_groups are required to set sync_status=FAIL, but 6 valid code_groups are required to set sync_status=OK, thus link timer is (6-4)*8ns longer than the duration of the invalid code stream.
e) The time measured in step (12) should establish an upper bound on the link_timer value used from the AN_RESTART state to the ABILITY_DETECT state.

**Possible Problems:** If a Next Page exchange is not supported or desired by the device, then the link_timer value used from the COMPLETE_ACKNOWLEDGE state to the NEXT_PAGE_WAIT state can not be observed.

**#37.2.3 Break Link Content**

**Purpose:** To verify that the DUT transmits Break Link containing rx_Config_Reg<D15:D0>=0 prior to entering ABILITY_DETECT.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - Figure 37-6 Auto-Negotiation State Diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** March 22, 1998

**Discussion:** On power-up, a clause 37 auto-negotiating device attempts to transmit /C/ codes to its link partner. As the device's management entity may need to write the desired advertised abilities to the appropriate registers, there is, necessarily, a delay before these values will be represented in the outgoing /C/ codes. To prevent the possibility of the device's link partner negotiating with the initial contents of the /C/ codes, clause 37 auto-negotiating devices are required to source an easily identifiable pattern on link start or restart. This pattern is a Config_Reg of all zero and is informally referred to as 'break link'. Thus, if a device needs to restart auto-negotiation for any reason, the transmission of break link ensures that the device's link partner will also restart.
This test verifies that the DUT sources a Config_Reg of all zero on startup, and on restart.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**
1. Power on the DUT.
2. Do not provide a link of any kind to the DUT.
3. Observe transmissions from the DUT.
4. Provide a link of /C/ codes with a Config_Reg of all zero.
5. Observe transmissions from the DUT up to and including the transmission of a non-zero Config_Reg.
6. Complete auto-negotiation and then attempt to restart auto-negotiation either by transmitting sufficient invalid codes to the DUT, or via management control.
7. Observe transmissions from the DUT up to and including the transmission of a non-zero Config_Reg.

**Observable Results:**

a) On power up, and enabling of auto-negotiation, the DUT should source /C/ codes with a Config_Reg of all zero indefinitely while not receiving a link from a remote device.
b) Once a link is established, an_sync_status=OK, and after the link_timer expires, the DUT should cease transmitting break link and commence transmission of its advertised abilities.
c) On auto-negotiation restart, the DUT should source /C/ codes with a Config_Reg of all zero.

**Possible Problems:** None.

**#37.2.4 Next Page Bit**

**Purpose:** To verify that the DUT sets the Next Page (NP) bit while the DUT desires the NP exchange.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.5.3, Figure 37-6 Auto-Negotiation state diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** Every 16 bit Config_Reg value exchanged between link partners is referred to as a page. Once the initial "base page" has been exchanged, a device may elect to transmit a new 16 bit Config_Reg value to convey additional data to its link partner. These additional pages are referred to as "next pages". Two classes of next pages exist: Message Pages and Unformatted Pages. During the exchange of next pages, the NP bit plays a simple role. It is to have the value 1 if a station has further next pages to transmit, and 0 if it is transmitting its last page of information. Note that a next page exchange will not occur unless both link partners set NP to one in the base page. Once both a device and its link partner have transmitted all of their next pages, the next page process should end and a link should be established. If a device has no further next pages to send, but its link partner is indicating it has additional next pages to send (via the NP bit), then the device is required to send a predefined Null Message Page to its link partner until the link partner has completed transmission of all its next pages. If a device's link partner finishes transmitting Message or Unformatted pages before the device and is sending out null pages, then the device should go directly from its last Message or Unformatted page to a link. If both devices transmit their last message or unformatted page at the same time, they should both go straight to a link and no null pages should be sent.
This test is designed to verify that the device under test properly sets the NP bit throughout the next page process.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. If necessary, force the DUT to request a next page exchange, by setting NP=1 in its Base Page.
2. Initialize by restarting auto-negotiation by any means.
3. Provide a stream of /C/ codes to the DUT such that it enters the COMPLETE_ACKNOWLEDGE state. The NP bit in the Config_Reg sent to the DUT should be 1.
4. Provide a stream of /C/ codes to the DUT, with bit D11 toggled from the previous Config_Reg value, until the DUT reenters the COMPLETE_ACKNOWLEDGE state.
5. Repeat step (4) until the DUT sets NP=0.

**Observable Results:**

a) The DUT should keep NP=1 for all pages exchanged, excluding the final page the DUT desires to transmit.
b) Once the DUT completes transmitting its last desired next page with NP=0, it should begin transmitting next pages containing the Null Message code (Config_Reg<D10:0>=00000000001b).

**Possible Problems:** If the DUT cannot be forced to send next pages (NP=1 in Base Page), then this test cannot be performed.

**#37.2.5 Null Message Page**

**Purpose:** To verify that the DUT properly sends Null Message Pages when the DUT has no further next pages to exchange.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.4.3 Next page function, Figure 37-6 Auto-Negotiation state diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** Once a device has finished sending Message and Unformatted next pages, it is required to send Null message pages until its link partner is done as well. Alternatively, if a device has no next pages to send, but it's link partner does, the device is required to send Null message pages until the link partner is done. A null message page is defined as a Message Page that contains the null message code 00000000001 (bits M[10:0]). This test is designed to verify that once the device under test completes its transmission of Message and Unformatted next pages, it sends out valid Null message pages until its link partner is done as well.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. If necessary, force the DUT to a mode where it will request a next page exchange, by setting NP=1 in its Base Page.
2. Initialize be restarting auto-negotiation by any means.
3. Provide a stream of /C/ codes to the DUT such that it enters the COMPLETE_ACKNOWLEDGE state. The NP bit in the Config_Reg sent to the DUT should be 1.
4. Provide a stream of /C/ codes to the DUT, with bit D11 toggled from the previous Config_Reg value, until the DUT reenters the COMPLETE_ACKNOWLEDGE state.
5. Repeat step (4) until the DUT sets NP=0, at which time, repeat step (4) three additional times, setting NP=0 the final time.
6. Once the DUT has entered COMPLETE_ACKNOWLEDGE, provide a stream of /I/ codes to the DUT.

**Observable Results:**

a) The DUT should source three Null message pages with NP=0, prior to sourcing /I/ codes.

**Possible Problems:** If the DUT cannot be forced to send next pages (NP=1 in Base Page), then this test cannot be performed.

#### #37.2.6 Toggle Bit

**Purpose:** To verify that the DUT properly sets the toggle bit (T) in all transmitted next pages.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.4.3, 37.2.4.3.6.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** During next page exchange, the toggle bit (D11) of the transmitted next pages serves a simple purpose. The value of this bit alternates between 0 and 1 in consecutive pages, and therefore provides the receiving station with a quick and easy check to verify that it is receiving next pages in the proper order. Therefore, it is important that the DUT sets these bits properly, starting with its first next page, whose toggle bit value takes the opposite value of bit D11 in the device's Config_Reg.
This test is designed to verify that the device under test sets the toggle bit properly throughout the exchange of next pages.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by restarting auto-negotiation by any means.
2. Provide a stream of /C/ codes to the DUT such that it enters the COMPLETE_ACKNOWLEDGE state. The NP bit in the Config_Reg sent to the DUT should be 1.
3. Provide a stream of /C/ codes to the DUT, with bit D11 toggled from the previous Config_Reg value, until the DUT reenters the COMPLETE_ACKNOWLEDGE state.
4. Repeat step (4) until the DUT sets NP=0 and/or sufficient next pages have been observed from the DUT.

**Observable Results:**

a) The value of the toggle bit in the first next page should have the opposite value of bit D11 in the DUT's base page Config_Reg.
b) The value of the toggle bit of the next page transmitted by the DUT should always take the opposite value of the toggle bit of the previous next page.

**Possible Problems:** If the DUT cannot be forced to send next pages (NP=1 in Base Page), then this test cannot be performed.

.

**#37.2.7 Message Page and Unformatted Page Encoding**

**Purpose:** To verify that the DUT properly encodes Message and Unformatted pages.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.4.3.1 Next page encodings, subclause 37.2.4.3.4. Message page
- IEEE 802.3u June 1995 Edition - Annex 28C.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** There are two types of pages that a device may send as next pages: message or unformatted pages. Each type of page has its own formatting. Message pages contain an 11-bit message code field that can contain only those values defined in Annex 28C. Unformatted pages contain an 11-bit unformatted code field whose encoding is either vendor specific, or defined by the preceding Message page.

Message Page encoding:

| **LSB** | | | | | | | | | | | | | | | **MSB** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 |
| M0 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | T | Ack2 | MP | Ack | NP |

Unformatted Page encoding:

| **LSB** | | | | | | | | | | | | | | | **MSB** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 |
| U0 | U1 | U2 | U3 | U4 | U5 | U6 | U7 | U8 | U9 | U10 | T | Ack2 | MP | Ack | NP |

Note the MP bit, which is required to be set to 1 if the page is a message page, and 0 if it is an unformatted page. This test is designed to verify that the DUT's message pages contain an acceptable message code followed by the appropriate number of unformatted pages (see Annex 28C for the message codes and appropriate number of unformatted pages), and all pages have the MP bit set correctly.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. If necessary, force the DUT to request a next page exchange, by setting NP=1 in its Base Page.
2. Initialize be restarting auto-negotiation by any means.
3. Provide a stream of /C/ codes to the DUT such that it enters the COMPLETE_ACKNOWLEDGE state. The NP bit in the Config_Reg sent to the DUT should be 1.
4. Provide a stream of /C/ codes to the DUT, with bit D11 toggled from the previous Config_Reg value, until the DUT reenters the COMPLETE_ACKNOWLEDGE state.
5. Repeat step (4) until the DUT sets NP=0, at which point send one /C/ code sequence to the DUT with the NP bit set to 0..
6. Once the DUT has entered COMPLETE_ACKNOWLEDGE for the page transmitted with NP=0, provide a stream of /I/ codes to the DUT.

**Observable Results:**

a) All Message pages should have MP=1, and all Unformatted pages should have MP=0
b) All Message pages transmitted by the DUT should contain a valid message code and should be followed by the appropriate number of unformatted pages, as defined in Annex 28C.

**Possible Problems:** If the DUT cannot be forced to send next pages (NP=1 in Base Page), then this test cannot be performed.

.

**#37.3.1 Ability Match**

**Purpose:** To verify that the DUT enters the ACKNOWLEDGE_DETECT state upon reception of three consecutive and consistent Config_Reg values, ignoring the ACK bit.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.3.1.2 Functions, Figure 37-6 Auto-Negotiation state diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** The core of auto-negotiation is based on the proper exchange of a 16-bit value between a station and its link partner. Encoded in this 16-bit value are the device's capabilities. Upon receipt of these capabilities, the link partner learns the capabilities of the attached station and can configure the link optimally. If the 16-bit value is corrupted in any way, then the link established could be sub-optimal, to the point of being non-functional. To ensure the proper exchange of these 16-bit values, matching functions are employed. These functions require the reception of three identical and consecutive 16 bit values before their acceptance.
The ability_match function looks for three identical and consecutive rx_Config_Reg<D15:D0> values, ignoring bit 14 (ACK). Once these are received, ability_match is set to TRUE. Referring to figure 37-6, Auto-Negotiation State Diagram, this results in a transition from the ABILITY_DETECT to ACKNOWLEDGE_DETECT states (provided rx_Config_Reg<D15:D0>≠0). Upon entering ACKNOWLEDGE_DETECT, the device sets its ACK bit to 1, indicating it has successfully received its link partner's 16-bit value, and then waits for a similar acknowledgment from the link partner (refer to the acknowledge_match test). When ability_match is looking for three identical and consecutive rx_Config_Reg values, it naturally must keep count. If an /I/ code is received, this count should be reset. Likewise, the count is reset if the second or third rx_Config_Reg does not match the first rx_Config_Reg.
This test explores the ability_match function by ensuring the match count is reset properly, and that once the count reaches three, ability_match=TRUE, and the device sets its ACK bit to one in its transmitted Config_Reg.

Note: Detection of ability_match=TRUE, while in the ABILITY_DETECT state, typically results in the transition to the ACKNOWLEDGE_DETECT, where 'transmit_ack=TRUE'. This results in the DUT transmitting a Config_Reg with bit 14 (ACK bit) set to 1. Thus, this process is readily observable, and will henceforth be referred to as the device 'setting its ACK bit'.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by resetting the link by any means.
2. Provide a link of /C/ codes with a Config_Reg of all zero.
3. Observe transmissions from the DUT up to and beyond the transmission of a non-zero Config_Reg.
4. Provide a stream of /C/ codes where Config_Reg takes on any non-zero value. Observe transmissions from the DUT.
5. Repeat step (1) and (4) varying the Config_Reg<D15:0> value to all defined abilities, and some encodings that are reserved ex: (FFF0h).
6. Provide a stream of /C/ codes where two Config_Reg<D15:0> values (which are 1 bit different) are sent repeatedly. ex: (01E0h), (01A0h)
7. Observe transmissions from the DUT
8. Repeat step (6) and (7) varying the Config_Reg value such that only 1 bit is ever different, but all 16 bits are eventually tested, performing the test of bit 14 (ACK) last.
9. Restart auto-negotiation by any means.
10. Provide a stream of /I/ codes to the DUT. Continue until the DUT sources a non-zero Config_Reg value..
11. Source a stream of (x) consecutive and identical Config_Reg values followed by (y) /I/ codes followed by one more identical Config_Reg value, followed by a stream of /I/ codes. (x) is initially one, and (y) is initially 0. Observe transmissions from the DUT. ex: (x)=2, (y)=1 2x(01A0h), 1x(/I/), 1x(01A0h)
12. Repeat step (9) through (11), increasing (x) until the device sets its ACK bit.
13. Repeat step (12), with (x) reset to one, and (y) kept at one for all iterations.

**Observable Results:**

a) In step (3), the DUT should not set its ACK bit upon detection of ability_match= TRUE*rx_Config_Reg<D15:D0>=0
b) In step (4) and (5), the DUT should set its ACK bit in all cases.
c) In step (6),(7) and (8), the DUT should not set its ACK bit upon reception of inconsistent Config_Reg values, with the exception of Config_Reg values that differ only by bit 14 (ACK).
d) In step (12), the DUT should set its ACK bit after three identical and consecutive Config_Reg values are received, ie (x)=2.
e) In step (13), the DUT should set its ACK bit only after three identical and consecutive Config_Reg values are received, ie (x)=3.

**Possible Problems:** None.

### #37.3.2 Acknowledge Match

**Purpose:** To verify that the DUT enters the COMPLETE_ACKNOWLEDGE state upon reception of three consecutive and consistent Config_Reg values, including the ACK bit.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.3.1.2 Functions, Figure 37-6 Auto-Negotiation state diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** The core of auto-negotiation is based on the proper exchange of a 16-bit value between a station and its link partner. Encoded in this 16-bit value are the device's capabilities. Upon receipt of these capabilities, the link partner learns the capabilities of the attached station and can configure the link optimally. If the 16-bit value is corrupted in any way, then the link established could be sub-optimal, to the point of being non-functional. To ensure the proper exchange of these 16-bit values, matching functions are employed. These functions require the reception of three identical and consecutive 16 bit values before their acceptance.
The acknowledge_match function looks for three identical and consecutive rx_Config_Reg<D15:D0> values, that have bit 14 (ACK) set to one. Once these are received, acknowledge_match is set to TRUE. Referring to figure 37-6, Auto-Negotiation State Diagram, this results in a transition from the ACKNOWLEDGE_DETECT to COMPLETE_ACKNOWLEDGE states (provided rx_Config_Reg<D15:D0> is identical to the value that set ability_match=TRUE, refer to the consistency_match test). Upon entering COMPLETE_ACKNOWLEDGE, the device knows that its link partner has received its 16-bit value. However, the link partner must receive three identical and consecutive 16-bit values with the ACK bit set, in order to proceed to the COMPLETE_ACKNOWLEDGE state. To increase the likelihood of this occurring, the device (now in the COMPLETE_ACKNOWLEDGE state) continues to send its Config_Reg value with the ACK bit set for a duration of 'link_timer'. When acknowledge_match is looking for three identical and consecutive rx_Config_Reg values with the ACK bit set, it naturally must keep count. If an /I/ code is received, this count should be reset. Likewise, the count is reset if the second or third rx_Config_Reg does not match the first rx_Config_Reg.
This test explores the acknowledge_match function by ensuring the match count is reset properly, and that once the count reaches three, acknowledge_match=TRUE, and the device proceeds to the COMPLETE_ACKNOWLEDGE state.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by resetting the link by any means.
2. Provide a repeating stream of at least 20 /C/ codes where Config_Reg<D15:0> takes on any value, so long as ACK=0 followed by one Config_Reg value with ACK=1. ex: 20x(01A0h),1x(41A0h)
3. Observe transmissions from the DUT.
4. Repeat step (1) through (3) as necessary, modifying (2) by increasing the number(x) of consecutive Config_Reg values sent with the ACK bit set. Observe transmissions from the DUT.
5. With the number(x) found in (4), repeat (1) through (3) altering the stream of /C/ codes such that two Config_Reg values with ACK=1 (which are 1 bit different) are sent (x) times.
   Observe transmissions from the DUT. ex: (x)=3; 20x(01A0h), 3x(41A0h, 4120h),
6. Repeat step (5) varying the Config_Reg value such that only 1 bit is ever different, but all 16 bits are eventually tested.
7. Restart auto-negotiation by any means.
8. Provide a repeating stream of at least 20 /C/ codes where Config_Reg<D15:0> takes on any value, so long as ACK=0, followed by (z) Config_Reg value(s) with ACK=1 followed by one /I/ code followed by one more identical Config_Reg value with ACK=1. (z) is initially 1. Observe transmissions from the DUT. ex: 20x(01A0h), 2x(41A0h), /I/, 1x(41A0h).
9. Repeat step (7) and (8), increasing (z) until the device sets its ACK bit.

**Observable Results:**

a) In step (3) and (4), the DUT should enter COMPLETE_ACKNOWLEDGE once (x) exceeds two.
b) In step (5) and (6), the DUT should not enter COMPLETE_ACKNOWLEDGE upon reception of inconsistent Config_Reg values.
c) In step (8), the DUT should enter COMPLETE_ACKNOWLEDGE once (z) exceeds two.

Note: Entrance into COMPLETE_ACKNOWLEDGE can be observed two ways:

1) By observing transmission of /I/ codes commencing ‘link_timer’ after the test.
2) By observing transmission of a next page from the DUT, ‘link_timer’ after the test.

**Possible Problems:** None.

**#37.3.3 Consistency Match**

**Purpose:** To verify that the DUT performs a consistency match test based on results of the ability_match and acknowledge_match function.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.3.1.2 Functions, Figure 37-6 Auto-Negotiation state diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** The core of auto-negotiation is based on the proper exchange of a 16-bit value between a station and its link partner. Encoded in this 16-bit value are the device's capabilities. Upon receipt of these capabilities, the link partner learns the capabilities of the attached station and can configure the link optimally. If the 16-bit value is corrupted in any way, then the link established could be sub-optimal, to the point of being non-functional. To ensure the proper exchange of these 16-bit values, matching functions are employed. These functions require the reception of three identical, and consecutive 16 bit values before their acceptance.
The consistency_match function takes the rx_Config_Reg<D15:D0> value that caused ability_match=TRUE while the device was in the ABILITY_DETECT state, and compares it with the rx_Config_Reg<D15:D0> value that caused acknowledge_match=TRUE. If they are identical (ignoring bit 14 {ACK}) then consistency_match is set to TRUE. If for any reason this is not the case, the consistency_match is set to FALSE. This prevents the device from altering the Config_Reg that was acknowledged in the ABILITY_DETECT state in any way, except for the setting of the ACK bit. Additionally, this prevents the (unlikely) event that three consecutive Config_Reg values could be received in error, but otherwise identical. If inconsistent values are found, auto-negotiation is restarted.
This test explores the consistency_match function by ensuring the match is performed on the proper Config_Reg values, and that auto-negotiation is properly restarted in the event of a consistency_match failure.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by resetting the link by any means.
2. Provide a repeating stream of at least 20 /C/ codes where Config_Reg<D15:0> takes on any value, so long as ACK=0, followed by (x) Config_Reg values with ACK=1. Where (x) is the value found in step (4) of test #37-3-2, and where the Config_Reg value is 1 bit different from the Config_Reg value when ACK=0. Observe transmissions from the DUT. ex: (x)=3; 20x(01A0h), 3x(4120h)
3. Repeat steps (1) and (2), in step (2) vary the Config_Reg value such that only 1 bit is ever different, but all 15 bits (the ACK must be set properly) are eventually tested.
4. Restart auto-negotiation by any means.
5. Provide a stream of /I/ codes to the DUT. Continue until the DUT sources a non-zero Config_Reg.
6. Source (x)*(y) identical and consecutive Config_Reg values (CR1_NoA) with ACK=0 to the DUT (causing ability_match=TRUE), then source a different stream of Config_Reg values (CR2_NoA) with ACK=0 (send at least (x)*(y)). Now alter the transmitted stream of /C/ codes such that the ACK bit is set, source a sufficient number such that acknowledge_match=TRUE. (current stream should be CR2_A). Follow the /C/ codes with a stream of /I/ codes. Observe transmissions from the DUT. ex: 3x(01A0h), 3x(01E0h), 3x(41E0h). (y) is initially 1. If problems are encountered, (y) is increased to 10 and this step is repeated.
7. Restart auto-negotiation by any means.
8. Provide a stream of /I/ codes to the DUT. Continue until the DUT sources a non-zero Config_Reg.
9. Source three identical and consecutive Config_Reg values with ACK=1 to the DUT (causing ability_match=TRUE, acknowledge_match=TRUE, and consistency_match=TRUE.) Follow the /C/ codes with a stream of /I/ codes. Observe transmissions from the DUT.

**Observable Results:**

a) In steps (2) and (3), the DUT should restart auto-negotiation due to a consistency_match failure.
b) In step (6), the DUT should restart auto-negotiation, due to a consistency_match failure, as ability_match was set based on CR1_NoA, and acknowledge_match was set on CR2_A (even though ability_match was true due to CR2_NoA at the time that acknowledge_match was set).
c) In step (9), the DUT should complete auto-negotiation and establish a link.

**Possible Problems:** None.

#### #37.3.4 Idle Match

**Purpose:** To verify that the DUT enters the LINK_OK state upon reception of three consecutive /I/ ordered sets and expiration of link_timer.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 36.2.4.12 IDLE (/I/), subclause 37.3.1.2 Functions, Figure 37-6 Auto-Negotiation state diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** April 28, 1998

**Discussion:** The core of auto-negotiation is based on the proper exchange of a 16-bit value between a station and its link partner. Once these values have been exchanged, the device must compare its link partner's abilities with its own, and resolve the most optimal link. This process of Priority Resolution is not instantaneous. Referring to Figure 37-6, the IDLE_DETECT state exists primarily to allow for a delay such that the proper link can be configured, hence the link_timer is used. A device is not permitted to proceed to the LINK_OK state unless the link_timer has expired. However, it is also required that an idle_match=TRUE occur. In order to be true, the idle_match function must receive three consecutive idle codes. As per subclause 36.2.4.12, any two code-groups beginning with the code-group K28.5, followed by a data code-group other than D21.5 or D2.2 should be treated as an idle code. Once idle_match=TRUE, the device knows that its link partner has exited the COMPLETE_ACKNOWLEDGE state and entered IDLE_DETECT, thus preventing a link from being established before the link partner has completed transmission of /C/ codes. This check is essential as reception of /C/ codes once the link is up would result in the auto-negotiation process being restarted. Like the ability_match and acknowledge_match functions, when idle_match is looking for three consecutive idle codes, it naturally must keep count. If a /C/ code is received, this count should be reset.
This test explores the idle_match function by ensuring the match count is reset properly, and that once the count reaches three, idle_match=TRUE, and the device proceeds to the LINK_OK state.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by resetting the device by any means.
2. Provide a stream of at least 20 /C/ codes where Config_Reg takes on any value, so long as ACK=0 followed by at least 10 Config_Reg values with the ACK bit set.
3. Provide a stream of /I/ codes to the DUT as specified in the note below.
4. Continuously send the streams defined in steps (2) and (3).
5. Observe transmissions from the DUT.
6. Repeat steps (1) through (5) until all test patterns have been sent.

Note - Test patterns include (but need not be limited to):

a) /-K28.5/+D16.2/-K28.5/+D16.2/-K28.5/+D16.2 …
b) /+K28.5/-D16.2/+K28.5/-D16.2 …
c) /+K28.5/-D5.6/-K28.5/+D5.6/+K28.5/-D5.6 …
d) /+K28.5/-D5.6/-K28.5/+D16.2/-K28.5/+D5.6/+K28.5/-D16.2 …
e) /-K28.5/+D16.2/-K28.5/+D10.2/+K28.5/-D16.2/+K28.5/-D10.2 …
f) /+K28.5/-D5.6/-K28.5/+D10.2/+K28.5/-D5.6/-K28.5/+D10.2 …
g) /-K28.5/+D10.2/+K28.5/-D10.2/+K28.5/-D10.2 …
h) /-K28.5/+D16.2/-K28.5/+D21.5/+K28.5/-D5.6 …
i) /-K28.5/+D16.2/-K28.5/+D2.2 …
j) /-K28.5/+D16.2/-K28.5/+D2.2/-D0.0/-D0.0/-K28.5/+D16.2/-K28.5/+D21.5/+D0.0/+D3.0 …
k) 2x(/-K28.5/+D16.2/) /-K28.5/+D2.2/-D0.0/-D0.0/ 2x(/-K28.5/+D16.2/) /-K28.5/+D21.5/+D0.0/+D3.0 …
l) 3x(/-K28.5/+D16.2/) /-K28.5/+D2.2/-D0.0/-D0.0/ 3x(/-K28.5/+D16.2/) /-K28.5/+D21.5/+D0.0/+D3.0 …

**Observable Results:**

a) Test patterns (a) through (g) should allow the DUT to enter the LINK_OK state.
b) Test patterns (h) through (k) should prevent the DUT from entering the LINK_OK state.
c) Test pattern (l) should allow the DUT to enter the LINK_OK state.

Note: Entrance into LINK_OK can be observed by:

- observing if the DUT restarts auto-negotiation upon receipt of a /C/ code. This should occur in the LINK_OK state, but not in the IDLE_DETECT state.
- if a non-zero Config_Reg does not restart auto-negotiation, then observing frame transmission from the DUT should indicate entrance into LINK_OK.

**Possible Problems:** None.

**#37.3.5 Reception of NP Bit**

**Purpose:** To verify that the DUT properly enters into a next page exchange upon receipt of NP=1.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.4.3 Next page function, 37.3.1.1 Variables, Figure 37-6 Auto-Negotiation state diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** March 24, 1998

**Discussion:** During the exchange of next pages, the NP bit plays a simple role. It is to have the value 1 if a station has further next pages to transmit, and 0 if it is transmitting its last page of information. If a device receives a base page with NP set to 1, it should enter into a next page exchange only if the device itself desires a next page exchange (indicated by mr_np_able=TRUE and NP=1 in base page Config_Reg). Thus, if a device supports next page exchange, but has no next page information to exchange, it may do one of two things. The device may set NP=1 in its base page and transmit next pages containing the Null Message code with NP=0 until its link partner has completed transmission of next pages. Or, the device may set NP=0 in its base page and upon receipt of next pages with NP=1, the device may change the value of the NP bit to 1 in its base page and restart the auto-negotiation process, and then proceed as described above. In this way, a device that has no next page information to exchange, can still acquire the next page information from its link partner.
Once both a device and its link partner have transmitted all of their next pages, the next page process should end and a link should be established. If a device's link partner finishes transmitting message or unformatted pages before the device and is sending out null pages, then the device should go directly from its last message or unformatted page to a link. If both devices transmit their last message or unformatted page at the same time, they should both go straight to a link and no null pages should be sent.
This test is designed to verify that the device under test properly enters into a next page exchange upon receipt of a base page with NP set to 1, provided the device supports and desires a next page exchange.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Force the DUT to request a next page exchange, by setting NP=1 in its Base Page.
2. Initialize by restarting auto-negotiation by any means.
3. Provide a stream of /C/ codes to the DUT such that it enters the COMPLETE_ACKNOWLEDGE state. The NP bit in the Config_Reg sent to the DUT should be 1.
4. Provide a stream of /C/ codes to the DUT, with bit D11 toggled from the previous Config_Reg value.
5. Observe transmissions from the DUT
6. Repeat steps (2) through (5), setting the NP bit to 0 in the Config_Reg sent to the DUT.
7. Force the DUT to bypass a next page exchange, by setting NP=0 in its Base Page.
8. Repeat steps (2) through (5).

**Observable Results:**

a) When the DUT has NP=1, it should enter into a next page exchange only when the NP bit is set in the Config_Reg received from the link partner.
b) When the DUT has NP=0, it should not enter into a next page exchange.

**Possible Problems:** If the DUT's NP bit cannot be set to 1 in the Base Page, then a next page exchange should not occur.

**#37.3.6 Reception of RF Bits**

**Purpose:** To observe the behavior of the DUT upon receipt of non-zero Remote Fault (RF) bits, and verify that the RF bit (1.4) is set in management.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.1.5 Remote fault.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** The purpose of auto-negotiation is to resolve the optimal link between link partners. If a problem is detected, either with the signaling received by a device, or with the device's transmitter, a device has the option of using the Remote Fault function to signal the problem to it's link partner. While its link partner may not implement the Remote Fault function, it is required to, at least, set bit 1.4 (Remote Fault) in the GMII management register set upon receipt of non-zero remote fault bits.
This test observes the device's reaction to the reception of configuration ordered sets with the RF bits set. Transmissions from the device on the medium will be monitored, as well as bit 1.4 of the management registers when possible.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize be restarting auto-negotiation by any means.
2. Provide a stream of /C/ and /I/ codes to the DUT such that it enters the LINK_OK state. The RF bits in the Config_Reg sent to the DUT should be 01.
3. Observe transmissions from the DUT, and bit 1.4 of the management registers.
4. Repeat steps (1) through (3) until the RF bit values of 10, and 11 have also been tested.

**Observable Results:**

a) The DUT should properly complete the auto-negotiation process.
b) The DUT should set bit 1.4 of the management registers.

**Possible Problems:** If access to the management registers is not possible, then bit 1.4 cannot be monitored.

**#37.3.7 Reception of Reserved Bits**

**Purpose:** To observe the behavior of the DUT upon receipt of non-zero Reserved bits.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - Table 37-5 AN advertisement register bit definitions, and Table 37-6 AN link partner ability base page register bit definitions.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** Within the 16-bit base page value are reserved bits which should be sent as zero and ignored upon receipt, as described in the Table 37-5: AN advertisement register bit definitions and Table 37-6: AN link partner ability base page register bit definition. Thus, if a future modification to the standard made use of these reserved bits, current devices should not treat the non-zero reception of these bits as an error.
This test observes the behavior of the device upon receipt of non-zero Reserved bits. As the values of these bits should be ignored when received, no irregular behavior should occur.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize be restarting auto-negotiation by any means.
2. Provide a stream of /C/ codes to the DUT such that it enters the IDLE_DETECT state. The reserved bits in the Config_Reg sent to the DUT should be 1. Observe transmissions from the DUT
3. Provide a stream of /I/ codes to the DUT and verify that a link is established.

**Observable Results:**

a) The DUT should complete negotiation and establish a link without error.

**Possible Problems:** None.

**#37.3.8 Reception of Toggle Bit**

**Purpose:** To verify the DUT checks the value of the toggle bit (T) when transitioning from NEXT_PAGE_WAIT to ACKNOWLEDGE_DETECT.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.4.3 Next page function, subclause 37.2.4.3.6 Toggle.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** During next page exchange, the toggle bit (D11) of the transmitted next pages serves a simple purpose. The value of this bit alternates between 0 and 1 in consecutive pages, and therefore provides the receiving station with a quick and easy check to verify that it is receiving next pages in the proper order. If a device receives consecutive pages without the toggle bit toggled, it should sit in the NEXT_PAGE_WAIT state until it receives a next page with the proper toggle bit value. This test is designed to verify that the DUT checks the value of the toggle bit in received next pages.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. If necessary, force the DUT to request a next page exchange, by setting NP=1 in its Base Page.
2. Initialize by restarting auto-negotiation by any means.
3. Provide a stream of /C/ codes to the DUT such that it enters the COMPLETE_ACKNOWLEDGE state. The NP bit in the Config_Reg sent to the DUT should be 1.
4. Provide a stream of /C/ codes to the DUT, with bit D11 the same as the previous Config_Reg value.
5. Provide a stream of /C/ codes to the DUT, with bit D11 toggled from the previous Config_Reg value, until the DUT reenters the COMPLETE_ACKNOWLEDGE state.
6. Repeat several combinations of steps (3) and (4), or steps (4) and (4).
7. Repeat steps (1) through (5), modifying step (3) such that bit D11 is initially toggled.

**Observable Results:**

a) At any time the DUT is receiving a new next page, with bit D11 untoggled, the DUT should not set its ACK bit in its outgoing Config_Reg.
b) Once the DUT receives a new next page, with bit D11 toggled, the DUT should set its ACK bit in its outgoing Config_Reg as per the ability_match function.

**Possible Problems:** If the DUT cannot be forced to send next pages (NP=1 in Base Page), then this test cannot be performed.

.

## #37.3.9 NP_RX Flag

**Purpose:** To verify the DUT properly stores the received Config_Reg NP value in the np_rx flag.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.3.1.1 Variables, Figure 37-6 Auto-Negotiation state machine.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** Next page exchange is initiated when both link partners set the NP bit to 1 in the initial Config_Reg (base page) exchanged. When a device has no remaining pages to transmit, it sets NP to 0. Referring to Figure 37-6: Auto-Negotiation state diagram, reception of a Config_Reg with NP set to 0 causes NP_RX=0 to be set in the COMPLETE_ACKNOWLEDGE state. Once 'link_timer' is done, the state machine then proceeds to the IDLE_DETECT state. The NP_RX flag is necessary as it is possible that the Config_Reg value received upon entrance to COMPLETE_ACKNOWLEDGE will not be the same as the Config_Reg value being received 'link_timer' later. Specifically, when the receiving device no longer has next pages to send, it will be sending next pages with NP set to 0. It's link partner may continue to send next pages and indicate so by setting NP to 1. Upon receiving the second to last next page from its link partner, the receiving device will enter the COMPLETE_ACKNOWLEDGE state. If, for whatever reason, the link partner then enters the NEXT_PAGE_WAIT state and starts sending its final next page (with NP=0), while the receiving device is still in COMPLETE_ACKNOWLEDGE (as it's 'link_timer' has not expired), then the receiving device will have initially seen a NP value of 1, which changed to 0 before it had exited COMPLETE_ACKNOWLEDGE. If the receiving device checked the NP value of the most recently received next page when 'link_timer' expired, then it would prematurely enter the IDLE_DETECT state, and both link partner's state machines would dead-lock.
This test verifies that the DUT checks the value of the NP bit upon entrance to COMPLETE_ACKNOWLEDGE, stores it (in NP_RX), and properly completes the next page exchange.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. If necessary, force the DUT to request a next page exchange, by setting NP=1 in its Base Page.
2. Initialize by restart auto-negotiation by any means.
3. Provide a stream of /C/ codes to the DUT such that it enters the COMPLETE_ACKNOWLEDGE state. The NP bit in the Config_Reg sent to the DUT should be 1.
4. Provide a stream of /C/ codes to the DUT, with bit D11 toggled from the previous Config_Reg value, until the DUT reenters the COMPLETE_ACKNOWLEDGE state.
5. Repeat step (3) at least one time beyond when the DUT sets NP=0.
6. Continue sending valid next pages to the DUT. The second to last next page should be sent a minimum number of times (ideally, the page need only be sent three times, provided ACK=1 and NP=1).
7. Provide the last next page (with NP=0) to the DUT, this should be sent long enough such that the device can exit the COMPLETE_ACKNOWLEDGE state from step (5), and properly complete the last page exchange.
8. When the DUT sources idle codes, provide a stream of idle codes, and verify that a link is established.

**Observable Results:**

a) The DUT should complete the proper exchange of all pages sent.

**Possible Problems:** If the DUT cannot be forced to send next pages (NP=1 in Base Page), then this test cannot be performed.

.

## #37.3.10 Message Page and Unformatted Page Handling

**Purpose:** To verify that the DUT properly receives Message and Unformatted pages.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.5.3.1 Next page encodings, subclause 37.2.5.3.4 Message page.
- IEEE 802.3u June 1995 Edition - Annex 28C.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** There are two types of pages that a device may receive as next pages: message or unformatted pages. Each type of page has its own formatting. Message pages contain an 11-bit message code field that can contain only those values defined in Annex 28C. Unformatted pages contain an 11-bit unformatted code field whose encoding is either vendor specific, or defined by the preceding Message page.

Message Page encoding:

| **LSB** | | | | | | | | | | | | | | | **MSB** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 |
| M0 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | T | Ack2 | MP | Ack | NP |

Unformatted Page encoding:

| **LSB** | | | | | | | | | | | | | | | **MSB** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 |
| U0 | U1 | U2 | U3 | U4 | U5 | U6 | U7 | U8 | U9 | U10 | T | Ack2 | MP | Ack | NP |

Note the MP bit, which is required to be set to 1 if the page is a message page, and 0 if it is an unformatted page.

This test is designed to verify that the DUT properly receives any combination/order of Message and Unformatted pages, provided the requirements of Annex 28C are met.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. If necessary, force the DUT to request a next page exchange, by setting NP=1 in its Base Page.
2. Initialize be restarting auto-negotiation by any means.
3. Provide a stream of /C/ codes to the DUT such that it enters the COMPLETE_ACKNOWLEDGE state. The NP bit in the Config_Reg sent to the DUT should be 1. (Note: when testing Message code #4, RF1,2 ≠ 00)
4. Provide a stream of /C/ codes to the DUT, with bit D11 toggled from the previous Config_Reg value, until the DUT reenters the COMPLETE_ACKNOWLEDGE state.
5. Repeat step (4) until the DUT sets NP=0, and test scenario is complete (see Note below).
6. Once the DUT has entered COMPLETE_ACKNOWLEDGE for the page transmitted with NP=0, provide a stream of /I/ codes to the DUT.
7. Repeat 1 through 5 until all test scenarios are complete (see Note below).

Note: The contents of the Config_Reg transmitted to the DUT in steps (3) and (4) should include:

- Test reaction to all defined Message codes (#1 to #6), followed by the proper number of Unformatted pages.
- Test reaction to several arbitrary Unformatted pages (#0, #7, #100, #2047).
- Test reaction to several reserved Message codes (#0, #7, #100, #2047).
- Test reaction to Message codes (#2 to #6) that should be followed by Unformatted pages but are not.
- Test reaction to all defined Message codes (#1 to #6) sent in Unformatted pages.

**Observable Results:**

a) All Message/Unformatted pages should be received by the device following the typical base-page exchange protocol as per the Auto-Negotiation state diagram.
b) Upon receipt of Message code #4, followed by an unformatted code of #0, the DUT may provided a Remote Fault indication in its transmitted Config_Reg.
c) If observable (either at the MDI or via management), Message codes sent without MP=1, should have no effect.

**Possible Problems:** If the DUT cannot be forced to send next pages (NP=1 in Base Page), then this test cannot be performed.

**#37.4.1 Full/Half Duplex Resolution**

**Purpose:** To verify that the DUT properly configures the highest common denominator duplex mode.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.4.2 Priority resolution function, subclause 37.2.1.2 Full Duplex, 37.2.1.3 Half Duplex, Figure 37-2 Config_Reg base page encoding.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** Once a station has successfully received it's link partner's abilities, it must resolve the optimal link to be established, based on its own capabilities and those received from the link partner. One mode of operation that must be resolved is the duplex mode of the link. Within the sixteen bit Config_Reg<D0:D15>, bit D5 is set to indicate support for Full duplex mode, and bit D6 is set to indicate support for Half duplex mode. Full duplex links have priority over half duplex.
This test is designed to verify that a device capable of both full and half duplex operation properly establishes a full duplex link when connected to a link partner advertising such capabilities. Additionally, the device should not resolve to a duplex mode which it does not support.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**
1. Initialize by restarting auto-negotiation by any means.
2. Provide a stream of /C/ codes to the DUT such that it enters the IDLE_DETECT state. In the Config_Reg, the full and half duplex bits <D5,D6> sent to the DUT should be 0b11. Observe transmissions from the DUT.
3. Provide a stream of /I/ codes to the DUT and verify that the proper duplex link is established by causing the DUT to simultaneously transmit and receive a packet. If any collision fragments were transmitted by the DUT, then it is in Half duplex mode.
4. Repeat (1) through (3), modifying step (2) such that bit values of 0b01, 0b10, and 0b00 are tested.

**Observable Results:**
a) If the DUT advertises both full and half duplex capability, then it should resolve to a full duplex mode in all cases where the FD bit received is set.
b) If the device does not support one of the duplex modes, then it should not establish a link when only that unsupported mode is advertised to the DUT.
c) When the device receives a Config_Reg with bits <D5,D6>=0b00, no link should be permanently established.

**Possible Problems:** None.

**#37.4.2 Pause Mode Resolution**

**Purpose:** To verify that the DUT configures the proper Pause mode based on table 37-4.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.1.4 Pause, subclause 37.2.4.2 Priority resolution function , Table 37-4 Pause priority resolution.
- IEEE 802.3x/D3.1 December 16, 1996 - annex 31B

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** Once a station has successfully received it's link partner's abilities, it must resolve the optimal link to be established, based on its own capabilities and those received from the link partner. One mode of operation that must be resolved is the flow control (pause) mode of the link. As per Annex 31B, flow control (pause) shall only be used on full duplex links. When a device has PAUSE Receive enabled, then reception of an 802.3x Pause frame shall cause the DUT to cease packet transmission for the duration of time specified in the Pause frame. When a device has PAUSE Transmit enabled, then it may source Pause frames to the link partner anytime the DUT wishes to pause the transmission of packets to it from the link partner. On a full duplex link, the pause mode to resolve is governed by the device's PAUSE and ASM_DIR (asymmetric direction) bits (<D7,D8>) in its transmitted Config_Reg<D0:D15>, as well as the value of these bits received from its link partner. These four bits then combine to form several possible scenarios, as listed in Table 37-4 and reproduced here for convenience.

Table 37-4—Pause priority resolution

| Local Device | | Link Partner | | Local Resolution | Link Partner Resolution |
|---|---|---|---|---|---|
| PAUSE | ASM_DIR | PAUSE | ASM_DIR | | |
| 0 | 0 | Don't Care | Don't Care | Disable PAUSE Transmit and Receive | Disable PAUSE Transmit and Receive |
| 0 | 1 | 0 | Don't Care | Disable PAUSE Transmit and Receive | Disable PAUSE Transmit and Receive |
| 0 | 1 | 1 | 0 | Disable PAUSE Transmit and Receive | Disable PAUSE Transmit and Receive |
| 0 | 1 | 1 | 1 | Enable PAUSE transmit, Disable PAUSE receive | Enable PAUSE receive, Disable PAUSE transmit |
| 1 | 0 | 0 | Don't Care | Disable PAUSE Transmit and Receive | Disable PAUSE Transmit and Receive |
| 1 | 0 | 1 | Don't Care | Enable PAUSE Transmit and Receive | Enable PAUSE Transmit and Receive |
| 1 | 1 | 0 | 0 | Disable PAUSE Transmit and Receive | Disable PAUSE Transmit and Receive |
| 1 | 1 | 0 | 1 | Enable PAUSE receive, Disable PAUSE transmit | Enable PAUSE transmit, Disable PAUSE receive |
| 1 | 1 | 1 | Don't Care | Enable PAUSE Transmit and Receive | Enable PAUSE Transmit and Receive |

This test is designed to verify that a device capable of flow control (pause) operation only does so on full duplex links, and resolves link operation in accordance with table 37-4.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by restarting auto-negotiation by any means.
2. Provide a stream of /C/ codes to the DUT such that it enters the IDLE_DETECT state. The full duplex bit in the Config_Reg sent to the DUT should be 1, and the PAUSE and ASM_DIR bits should be 11. Observe transmissions from the DUT
3. Provide a stream of /I/ codes to the DUT and verify that the proper pause mode is established. This can be done by transmitting pause frames to the DUT, and also by transmitting any type of frames at a high rate to the DUT.
4. Repeat (1) through (3), modifying step (2) such that bit values of 01, 10 and 00 are tested.
5. Repeat (4), modifying step (2) such that the full duplex bit is 0 and the half duplex bit is 1 in the Config_Reg sent to the DUT.

**Observable Results:**

a) Depending on the DUT's advertised pause capabilities, the proper pause mode should be resolved as per table 37-4. This can be verified by observing if the DUT ceases frame transmission upon receipt of pause frames, or generates pause frames when heavily loaded.

b) The DUT should not respond to or send pause frames when the link resolved is half duplex.

**Possible Problems:** It may be difficult to load a device's port to the point where it needs to pause the station transmitting to that port. In such a case, it may never generate a pause frame. Thus, in these cases, failure to observe pause transmissions in itself is not sufficient to conclude that the device has not resolved the proper pause transmit mode.

## #37.5.1 Reception of Break Link

**Purpose:** To verify that the DUT restarts auto-negotiation upon receipt of rx_Config_Reg<D15:D0>=0.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - Figure 37-6 Auto-Negotiation state diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 26, 1998

**Discussion:** On power-up, a clause 37 auto-negotiating device attempts to transmit /C/ codes to its link partner. As the device's management entity may need to write the desired advertised abilities to the appropriate registers, there is, necessarily, a delay before these values will be represented in the outgoing /C/ codes. To prevent the possibility of the device's link partner negotiating with the initial contents of the /C/ codes, clause 37 auto-negotiating devices are required to source an easily identifiable pattern on link start or restart. This pattern is a Config_Reg of all zero and is informally referred to as 'break link'. Thus, if a device is reset, or commanded to restart auto-negotiation, or losses link(sync), it will transmit break link, and its link partner should recognize this event and restart its auto-negotiation process.

This test verifies that the DUT restarts auto-negotiation upon receipt of a Config_Reg of all zero when not in the AN_ENABLE, AN_RESTART, or ABILITY_DETECT states.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by restarting auto-negotiation by any means.
2. Provide a link of /I/ codes such that the DUT enters the ABILITY_DETECT state.
3. Provide a link of /C/ codes with a Config_Reg of zero
4. Observe the DUT's response.
5. If necessary, restart auto-negotiation by any means.
6. Repeat steps (2) through (5), causing the device in step (2) to enter the ACKNOWLEDGE_DETECT, COMPLETE_ACKNOWLEDGE, NEXT_PAGE_WAIT, or IDLE_DETECT states.

Note:

- Acknowledge Detect can be entered by providing a link of /C/ codes with a non-zero Config_Reg, with ACK=0.
- Complete Acknowledge can be entered by providing a link of /C/ codes with a non-zero Config_Reg, with ACK=1.
- Next Page Wait can be entered by providing a link of /C/ codes with a non-zero Config_Reg with ACK=1 and NP=1 provided the NP bit in the DUT's transmitted Config_Reg is 1.
- Idle Detect is entered link_timer after entrance into Complete Acknowledge, provided NP=0 in the Config_Reg sent to the DUT.

**Observable Results:**

a) Reception of a Config_Reg of all zero while the DUT is in the ABILITY_DETECT state should not restart auto-negotiation
b) Reception of a Config_Reg of all zero while the DUT is in the ACKNOWLEDGE_DETECT, COMPLETE_ACKNOWLEDGE, NEXT_PAGE_WAIT, or IDLE_DETECT states should restart auto-negotiation, causing the DUT to transmit a zero Config_Reg.

**Possible Problems:** If the DUT cannot send next pages then the Next Page Wait test cannot be performed.

## #37.5.2 Restoration of Sync

**Purpose:** To verify that the DUT properly restarts auto-negotiation link_timer after sync_status=OK.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.3.1.1 Variables, subclause 37.3.1.4 Timers, figure 36-9 Synchronization state diagram, figure 37-6 Auto-Negotiation state diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 27, 1998

**Discussion:** Clause 37 Auto-Negotiation learns of a loss of signal or synchronization via the an_sync_status variable. This variable is set to false 'link_timer' after the Synchronization state diagram (Figure 36-9) sets sync_status=FAIL. Upon such an event, the auto-negotiation process should restart (as observed by the transmission of break link). However, so long as an_sync_status is false, then the auto-negotiation process never exits the AN_ENABLE state. Once the synchronization state machine sets sync_status=OK, an_sync_status should become true immediately, and the auto-negotiation state process should proceed to AN_RESTART. After 'link_timer' has expired, the device should be observed to transmit its base page abilities.
This test observes transmissions from the DUT to determine if it properly restarts auto-negotiation once synchronization is reestablished.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by restarting auto-negotiation by any means.
2. Complete auto-negotiation and provide a stream of /I/ codes to the DUT.
3. Now that the link is up, provide a stream of invalid codes such that sync_status=FAIL, and link_timer after reception begins, an_sync_status=FAIL. ex: a repeating pattern of /-K28.5/-D16.2/-K28.5/-D16.2/ …
4. Observe transmissions from the DUT.
5. The DUT should be sourcing a stream of /C/ codes with a zero Config_Reg.
6. Wait 10 seconds (an arbitrary period of time).
7. Provide a stream of valid /C/ codes with a Config_Reg of all zero to the DUT. Observe transmissions from the DUT.

**Observable Results:**

a) In step (7), the DUT should set an_sync_status=OK after six valid code-groups are received, and after link_timer expires, the DUT should commence transmission of a non-zero Config_Reg.

**Possible Problems:** None.

### #37.5.3 Reception of /C/ codes in LINK_OK

**Purpose:** To verify that the DUT restarts auto-negotiation upon receipt of /C/ codes while xmit=DATA.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - Figure 36-7a PCS receive state diagram, part a; Figure 36-7b PCS receive state diagram, part b; Figure 37-6 Auto-Negotiation State Diagram.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 27, 1998

**Discussion:** Once a clause 37 auto-negotiating device completes the auto-negotiation process and establishes a link, it sets the variable 'xmit' to DATA, signifying that packet exchange can commence. If, at any time thereafter, the device receives a stream of /C/ ordered sets such that an ability_match occurs, then the device is forced to cease packet exchange and restart the auto-negotiation process. This is necessary as the device's link partner could restart its auto-negotiation process for many reasons once the link is up.
This test verifies that the DUT restarts auto-negotiation upon the receipt of /C/ codes. The reception of /C/ codes in the middle of a frame should result in an early end indication prior to auto-negotiation restart (refer to figure 36-7b).

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by restarting auto-negotiation by any means.
2. Provide a link of /C/ codes such that the DUT enters the COMPLETE_ACKNOWLEDGE state.
3. Once the DUT sources /I/ codes, provide a stream of /I/ codes to the DUT.
4. Embed a request frame in the stream of /I/ codes to the DUT.
5. Once the DUT responds to the request frame, source a stream of valid /C/ codes to the DUT.
6. Observe transmissions from the DUT.
7. Repeat steps (1) to (6), modifying step (5) so that one /C/ code is embedded in the idle stream to the DUT.
8. Repeat step (7), each time increasing the number(x) of consecutive identical /C/ codes until the DUT restarts auto-negotiation. (provided such a response was observed in the original step 5).
9. With the number(x) found in step (8), repeat step (7) modifying the location of the /C/ codes such that the test sequence is embedded in a frame transmitted to the DUT.

**Observable Results:**

a) Step (5), and (9) should result in the DUT restarting auto-negotiation, as observed by the transmission of /C/ codes from the DUT.
b) Step (7) should not result in the DUT restarting auto-negotiation.
c) Step (8) should only result in the DUT restarting auto-negotiation once a sufficient number of /C/ codes have been received to cause ability_match=TRUE.

**Possible Problems:** None.

## #37.6.1 Possible Offline Indication

**Purpose:** To observe if the DUT uses the Remote Fault function to signal Offline prior to being removed from the active configuration.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.1.5 Remote fault, 37.2.1.5.2 Offline.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 27, 1998

**Discussion:** A device that elects to support the Remote Fault function may use the remote fault encoding of 0b01 in the transmitted Config_Reg to indicate that the station is about to go offline. This may occur prior to the station powering off, running transmitter tests, or removing itself from the active configuration.
This test observes the device's transmitted RF bits under these circumstances in order to determine if the device supports the offline indication.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by restarting auto-negotiation by any means.
2. Provide a stream of /C/ and /I/ codes to the DUT such that it enters the LINK_OK state.
3. If possible, force the DUT to run a transmitter test. Observe transmissions from the DUT. Repeat steps (1) and (2) as necessary.
4. If possible, remove the device from the active configuration (by unloading the driver, etc). Observe transmissions from the DUT. Repeat steps (1) and (2) as necessary.
5. Power down the DUT. Observe transmissions from the DUT.

**Observable Results:**

a) The DUT may transmit a Config_Reg with the Remote Fault bits set to 0b01 in response to the conditions in step (3), (4) and/or (5). This test is not judged on a Pass/Fail basis.

**Possible Problems:** None.

## #37.6.2 Possible Link_Failure Indication

**Purpose:** To observe if the DUT uses the Remote Fault function to indicate a previous link failure.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.1.5 Remote fault, 37.2.1.5.3 Link_Failure.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 27, 1998

**Discussion:** A device that elects to support the Remote Fault function may use the remote fault encoding of 0b10 in the transmitted Config_Reg to indicate it has previously detected a link failure. A link failure may be signaled when an_sync_status=FAIL. Once an_sync_status=TRUE the stored remote fault signaling may be transmitted.
This test observes the device's transmitted RF bits under these circumstances in order to determine if the device supports the Link_Failure indication.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by restarting auto-negotiation by any means.
2. Provide a stream of /C/ and /I/ codes to the DUT such that it enters the LINK_OK state.
3. Create a an_sync_status=FAIL condition, either via invalid code transmission, of loss of signal.
4. Provide a stream of /C/ codes to the DUT and observe transmissions from the DUT.

**Observable Results:**

a) The DUT may transmit a Config_Reg with the Remote Fault bits set to 0b10 in response to the conditions in step (3). This test is not judged on a Pass/Fail basis.

**Possible Problems:** None.

## #37.6.3 Auto_Negotiation Error

**Purpose:** To observe if the DUT uses the Remote Fault function to indicate that the link partners are incompatible.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.1.5 Remote fault, 37.2.1.5.4 Auto-Negotiation_Error

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 27, 1998

**Discussion:** A device that elects to support the Remote Fault function must use the remote fault encoding of 0b11 in the transmitted Config_Reg to indicate that auto-negotiation cannot resolve the link partners abilities to a compatible link. One possible scenario is a device that supports only full duplex operation, and a link partner that supports only half duplex operation.
This test observes the device's transmitted RF bits under these circumstances in order to determine if the device supports the Auto_Negotiation Error indication.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**
1. Initialize by restarting auto-negotiation by any means.
2. Provide a stream of /C/ and /I/ codes to the DUT such that it enters the IDLE_DETECT state. In the base page Config_Reg values sent to the DUT, advertise a mode of operation incompatible with the DUT's advertised abilities.
3. Observe transmissions from the DUT.

**Observable Results:**
a) If the DUT supports the Remote Fault Function, it should set the RF bits to 0b10 in response to the conditions in step (2).

**Possible Problems:** None.

## #37.6.4 RF Set Duration

**Purpose:** When the Remote Fault function is in use, this test verifies that RF remains set until the DUT enters the COMPLETE_ACKNOWLEDGE state.

**References:**

- IEEE 802.3z/D4.1 February 5, 1998 - subclause 37.2.1.5 Remote fault.

**Resource Requirements:**

- A testing station capable of transmitting (receiving) arbitrary ten-bit code_groups using the signaling method of clause 38 or 39.

**Last Modification:** February 27, 1998

**Discussion:** A device that elects to support the Remote Fault function must not reset the RF encoding until the device transitions to the COMPLETE_ACKNOWLEDGE state during base page exchange, thus ensuring that the link partner receives the fault indication.
This test observes the device's transmitted RF bits when signaling a remote fault, in order to determine if the device continues sending the RF code until the COMPLETE_ACKNOWLEDGE state is entered.

**Test Setup:** Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the proper medium (i.e. balanced cable, multi-mode fiber).

**Procedure:**

1. Initialize by restarting auto-negotiation by any means.
2. Recreate any scenario from tests 37.6.1, 37.6.2, or 37.6.3 that causes a remote fault indication, or manually force the DUT to transmit a remote fault indication via management.
3. Provide a stream of /C/ codes such that the DUT enters the COMPLETE_ACKNOWLEDGE state.
4. Observe transmissions from the DUT.

**Observable Results:**

a) If the DUT supports the Remote Fault Function, it should keep the RF bits set until the device enters the COMPLETE_ACKNOWLEDGE state while exchanging the base page.

**Possible Problems:** None.