IEEE 100BASE-T1 Physical Coding Sublayer Test Suite

Version 1.1

Author & Company	Curtis Donahue, UNH-IOL Stephen Johnson, UNH-IOL
Title	IEEE 100BASE-T1 Physical Coding Sublayer Test Suite
Version	1.1
Date	August 3, 2017
Status	Final
Restriction Level	Public

This suite of tests has been developed to help implementers evaluate the functionality of their 100BASE-T1 Physical Coding Sublayer (PCS) based products. This test suite covers both PCS Transmit and PCS Receive, including state diagrams, encoding/decoding, and scrambling/descrambling.

OPEN Alliance

Disclaimer

NOTICE TO USERS WHO ARE OPEN ALLIANCE SIG MEMBERS: Members of OPEN Alliance have the right to use and implement this Specification, subject to the Member's continued compliance with the OPEN Alliance SIG's goverance documents, Intellectual Property Rights Policy, and the applicable OPEN Alliance Promoter or Adopters Agreement. OPEN Specification documents may only be reproduced in electronic or paper form or utilized in order to achieve the Scope, as defined in the OPEN Alliance Intellectual Property Rights Policy. Reproduction or utilization for any other purposes as well as any modification of the Specification document, in any form or by any means, electronic or mechanical, including photocopying and microfilm, is explicitly excluded.

NOTICE TO NON-MEMBERS OF OPEN ALLIANCE SIG: If you are not a Member of OPEN Alliance and you have obtained a copy of this document, you only have a right to review this document for informational purposes. You do not have the right to reproduce, distribute, make derivative works of, publicly perform or publicly display this document in any way.

All OPEN Specifications are provided on an "as is" basis and all warranties, either explicit or implied, are excluded unless mandatory under law. Accordingly, the OPEN Alliance Members who have contributed to the OPEN Specifications make no representations or warranties with regard to the OPEN Specifications or the information (including any software) contained therein, including any warranties of merchantability, fitness for purpose, or absence of third party rights and make no representations as to the accuracy or completeness of the OPEN Specifications or any information contained therein.

The OPEN Alliance Members who have contributed to the OPEN Specifications will not be liable for any losses, costs, expenses or damages arising in any way out of use or reliance upon any OPEN Specification or any information therein. Nothing in this document operates to limit or exclude any liability for fraud or any other liability which is not permitted to be excluded or limited by operation of law.

The material contained in OPEN Specifications is protected by copyright and may be subject to other types of Intellectual Property Rights.

The distribution of OPEN Specifications shall not operate as an assignment or license to any recipient of any OPEN Specification of any patents, registered designs, unregistered designs, trademarks, trade names or other rights as may subsist in or be contained in or reproduced in any OPEN Specification. The commercial exploitation of the material in this document may require such a license, and any and all liability arising out of use without such a license is excluded.

Without prejudice to the foregoing, the OPEN Alliance Specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

OPEN Alliance reserves the right to withdraw, modify, or replace any OPEN Specification at any time, without notice.

Version Control of Document

Version	Author	Description	Date
1.0	Stephen Johnson	• Document updated to reference IEEE 802.3bw 100BASE-T1 Standard. • Updated observable result for Test 3.4.8 – Receive State Diagram - THIRD SSD State part b.	06/9/2016
1.1	Stephen Johnson	• Updated reference table • Updated references and procedure for Test 3.1.1 – PCS Signaling • Updated discussion and procedure for Test 3.1.2 – PCS Reset • Updated references for Test 3.1.3 – Transmit Proper SSD, Test 3.1.4 – Transmit Proper ESD, Test 3.1.5 – Transmit ESD with tx_error, Test 3.1.7 – tx_error, Test 3.5.1 – JAB State Diagram – rcv_max_timer	3/2/2017
1.0_Internal	Stephen Johnson	• Updated version number to 1.0_Internal • Removed UNH IOL disclaimer • Removed references to UNH IOL • Updated Appendix 3.C – Line Tap • Removed last updated section from each test	5/4/17
1.0	Stephen Johnson	• Draft finalized • Restriction level changed to Public	6/6/17
1.1	Stephen Johnson	• Removed second restriction level table	8/3/17

Restriction level history of Document

Version	Restriction Level	Description	Date
1.0	OPEN internal only		06/9/2016
1.1	OPEN internal only		3/2/2017
1.0_Internal	OPEN internal only		5/4/17
1.0	Public		6/6/17
1.1	Public		8/3/17

Contents

1 ACKNOWLEDGMENTS

The OPEN Alliance would like to acknowledge the efforts of the following individuals in the development of this test suite.

Curtis Donahue	University of New Hampshire
Dave Estes	University of New Hampshire
Stephen Johnson	University of New Hampshire
Matt Kilpeck	University of New Hampshire
Shawn King	University of New Hampshire
Jeff Lapak	University of New Hampshire
Matt Mayer	University of New Hampshire
Alex Seiger	University of New Hampshire

2 INTRODUCTION

This particular suite of tests has been developed to help implementers evaluate the functionality of the PCS sublayer of their 100BASE-T1 products.

These tests are designed to determine if a product conforms to specifications defined in IEEE 802.3bw 100BASE-T1 Standard. Successful completion of all tests contained in this suite does not guarantee that the tested device will operate with other devices. However, combined with satisfactory operation in the OPEN Alliance interoperability test bed, these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many 100BASE-T1 automotive environments.

The tests contained in this document are organized in such a manner as to simplify the identification of information related to a test, and to facilitate in the actual testing process. Tests are organized into groups, primarily in order to reduce setup time in the lab environment, however the different groups typically also tend to focus on specific aspects of device functionality. A three-part numbering system is used to organize the tests, where the first number indicates the section of Clause 96 of the IEEE 802.3bw Standard on which the test suite is based. The second and third numbers indicate the test's group number and test number within that group, respectively. This format allows for the addition of future tests to the appropriate groups without requiring the renumbering of the subsequent tests.

The test definitions themselves are intended to provide a high-level description of the motivation, resources, procedures, and methodologies pertinent to each test. Specifically, each test description consists of the following sections:

Purpose

The purpose is a brief statement outlining what the test attempts to achieve. The test is written at the functional level.

References

This section specifies source material *external* to the test suite, including specific subsections pertinent to the test definition, or any other references that might be helpful in understanding the test methodology and/or test results. External sources are always referenced by number when mentioned in the test description. Any other references not specified by number are stated with respect to the test suite document itself.

Resource Requirements

The requirements section specifies the test hardware and/or software needed to perform the test. This is generally expressed in terms of minimum requirements, however in some cases specific equipment manufacturer/model information may be provided.

Last Modification

This specifies the date of the last modification to this test.

Discussion

The discussion covers the assumptions made in the design or implementation of the test, as well as known limitations. Other items specific to the test are covered here.

Test Setup

The setup section describes the initial configuration of the test environment. Small changes in the configuration should not be included here, and are generally covered in the test procedure section, below.

Test Procedure

The procedure section of the test description contains the systematic instructions for carrying out the test. It provides a cookbook approach to testing, and may be interspersed with observable results.

Observable Results

This section lists the specific observables that can be examined by the tester in order to verify that the DUT is operating properly. When multiple values for an observable are possible, this section provides a short discussion on how to interpret them. The determination of a pass or fail outcome for a particular test is generally based on the successful (or unsuccessful) detection of a specific observable.

Possible Problems

This section contains a description of known issues with the test procedure, which may affect test results in certain situations. It may also refer the reader to test suite appendices and/or whitepapers that may provide more detail regarding these issues.

3 Reference Table

Test Name	Test Number	IEEE 802.3bw References
Group 1: PCS Transmit		
PCS Signaling	Test 3.1.1	• [1] IEEE Std. 802.3bw subclause 96.3.3.1 – 4B/3B conversion • [2] IEEE Std. 802.3bw subclause 96.3.3.3.1 – Side-stream scrambler polynomial • [3] IEEE Std. 802.3bw subclause 96.3.3.3.2 – Generation of Syn[2:0] • [4] IEEE Std. 802.3bw subclause 96.3.3.3.3 – Generation of Scn[2:0] • [5] IEEE Std. 802.3bw subclause 96.3.3.3.4 – Generation of scrambled bits Sdn[2:0] • [6] IEEE Std. 802.3bw subclause 96.3.3.3.5 – Generation of ternary pair (TA_n, TB_n) • [7] IEEE Std. 802.3bw table 96-1 – Idle symbol mapping in training • [8] IEEE Std. 802.3bw table 96-2 – Data symbols when tx_mode=SEND_N • [9] IEEE Std. 802.3bw table 96-3 – Idle symbols when tx_mode=SEND_N
PCS Reset	Test 3.1.2	• [1] IEEE Std. 802.3bw subclause 96.3.1 – PCS Reset function
Transmit Proper SSD	Test 3.1.3	• [1] IEEE Std. 802.3bw subclause 96.3.3 – PCS transmit • [2] IEEE Std. 802.3bw subclause 96.3.3.3.5 – Generation of ternary pair (TA_n, TB_n)
Transmit Proper ESD	Test 3.1.4	• [1] IEEE Std. 802.3bw subclause 96.3.3 – PCS transmit • [2] IEEE Std. 802.3bw subclause 96.3.3.3.5 – Generation of ternary pair (TA_n, TB_n)
Transmit ESD with tx_error	Test 3.1.5	• [1] IEEE Std. 802.3bw subclause 96.3.3 – PCS transmit • [2] IEEE Std. 802.3bw subclause 96.3.3.3.5 – Generation of ternary pair (TA_n, TB_n)
Transmission of Stuff Bits	Test 3.1.6	• [1] IEEE Std. 802.3bw subclause 96.3.3.1.2 – 4B/3B conversion for MII data
tx_error	Test 3.1.7	• [1] IEEE Std. 802.3bw subclause 96.3.3 – PCS transmit • [2] IEEE Std. 802.3bw Figure 96-6: 4B3B MII Signal Conversion
Group 2: PCS Transmit State Diagram		
Transmit State Diagram	Tests 3.2.1-3.2.10	• [1] IEEE Std. 802.3bw figure 96-7 – PCS Transmit state diagram
Group 3: PCS Receive		
Receive PCS Signaling	Test 3.3.1	• [1] IEEE Std. 802.3bw subclause 96.3.4 – PCS Receive
Automatic Polarity Detection	Test 3.3.2	• [1] IEEE Std. 802.3bw subclause 96.3.4.4 – PCS Receive automatic polarity detection (optional)
Receive SSD	Test 3.3.3	• [1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram • [1] IEEE Std. 802.3bw subclause 96.3.4.2 – PCS Receive symbol decoding
Receive ESD	Test 3.3.4	• [1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram • [1] IEEE Std. 802.3bw subclause 96.3.4.2 – PCS Receive symbol decoding
Receive ERR_ESD3	Test 3.3.5	• [1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram • [1] IEEE Std. 802.3bw subclause 96.3.4.2 – PCS Receive symbol decoding
Reception of Stuff Bits	Test 3.3.6	• [1] IEEE Std. 802.3bw subclause 96.3.4.2 – PCS Receive symbol decoding

De-Interleave Ternary Pairs	Test 3.3.7	• [1] IEEE std. 802.3bw subclause 96.3.4.2 PCS Receive symbol decoding
Group 4: PCS Receive State Diagram		
Receive State Diagram	Tests 3.4.1-3.4.13	• [1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram
Group 5: JAB State Diagram		
JAB State Diagram - rcv_max_timer	Test 3.5.1	• [1] IEEE Std. 802.3bw figure 96-11 – JAB state diagram • [1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram • [2] IEEE Std. 802.3bw subclause 96.3.4.1.3 – Timer

All tests are designed for the IEEE 802.3bw 100BASE-T1 Standard.

4 Device Under Test (DUT) Requirements

For the purposes of this test suite, a DUT is one port of a 100BASE-T1 capable device that includes a 100BASE-T1 PHY mounted on a PCB and an MDI connector. Please see the additional requirements listed in the following table:

Test Number and Name	**Required Capabilities**
General	The Ability to send and receive frames[1]
Test 3.1.2 – PCS Reset	Access to PCS Reset
Test 3.1.5 – Transmit ESD with tx_error	MII Access
Test 3.1.7 – tx_error	MII Access
GROUP 2: PCS Transmit State Diagram	MII Access
Test 3.3.2 – Automatic Polarity Detection (Optional)	The DUT must implement Automatic Polarity Detection
GROUP 4: PCS Receive State Diagram	MII Access
GROUP 5: JAB State Diagram	MII Access

1) This can be accomplished through a loopback, responding to ICMP requests, or by forwarding traffic through two ports.

5 GROUP 1: PCS Transmit

Overview:

The tests defined in this section verify the PCS Transmit process defined for 100BASE-T1 capable PHYs in sections 96.3.1 and 96.3.3 of the IEEE 802.3bw 100BASE-T1 Specification.

5.1 Test 3.1.1 – PCS Signaling

Purpose: To verify that the PCS properly performs the 4B/3B conversion, side-stream scrambling, and ternary symbol generation.

References:

[1] IEEE Std. 802.3bw subclause 96.3.3.1 – 4B/3B conversion
[2] IEEE Std. 802.3bw subclause 96.3.3.3.1 – Side-stream scrambler polynomial
[3] IEEE Std. 802.3bw subclause 96.3.3.3.2 – Generation of Syn[2:0]
[4] IEEE Std. 802.3bw subclause 96.3.3.3.3 – Generation of Scn[2:0]
[5] IEEE Std. 802.3bw subclause 96.3.3.3.4 – Generation of scrambled bits Sdn[2:0]
[6] IEEE Std. 802.3bw subclause 96.3.3.3.5 – Generation of ternary pair (TA_n, TB_n)
[7] IEEE Std. 802.3bw table 96-1 – Idle symbol mapping in training
[8] IEEE Std. 802.3bw table 96-2 – Data symbols when TXMODE=SEND_N
[9] IEEE Std. 802.3bw table 96-3 – Idle symbols when TXMODE=SEND_N

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:

The 100BASE-T1 PCS uses a combination of a 4B3B conversion for MII data, side-stream scrambling, and ternary symbols to encode the data for transmission on the line. In addition to the scrambler, there are additional functions performed on the data and idle streams to eliminate the correlation transmit data and to balance the power density.

This test is designed to ensure that the DUT properly performs the 100BAE-T1 encoding during training, idle transmission, and data transmission.

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Test Procedure:

1. Configure the DUT as master and the Link Partner as slave.
2. Monitor and decode the transmissions from the DUT while the DUT is in tx_mode = SEND_I.
3. Monitor and decode the transmissions from the DUT while the DUT is in tx_mode = SEND_N and tx_enable = 1.
4. Monitor and decode the transmissions from the DUT while the DUT is in tx_mode = SEND_N and tx_enable = 0.
5. Configure the DUT as slave and the Link Partner as master.
6. Monitor and decode the transmissions from the DUT while the DUT is in tx_mode = SEND_I.
7. Monitor and decode the transmissions from the DUT while the DUT is in tx_mode = SEND_N and tx_enable = 1.
8. Monitor and decode the transmissions from the DUT while the DUT is in tx_mode = SEND_N and tx_enable = 0.

Observable Results:

a. The transmissions in Step 2 should follow the 4B3B conversion specified in [1], the side-stream scrambler polynomial specified in [2] for master, the generation of Sdn[2:0] specified in [5], and the ternary symbol mapping defined in [6].
b. The transmissions in Step 3 should follow the 4B3B conversion specified in [1], the side-stream scrambler polynomial specified in [2] for master, the generation of Sdn[2:0] specified in [5], and the ternary symbol mapping defined in [8].
c. The transmissions in Step 4 should follow the 4B3B conversion specified in [1], the side-stream scrambler polynomial specified in [2] for master, the generation of Sdn[2:0] specified in [5], and the ternary symbol mapping defined in [9].
d. The transmissions in Step 6 should follow the 4B3B conversion specified in [1], the side-stream scrambler polynomial specified in [2] for master, the generation of Sdn[2:0] specified in [5], and the ternary symbol mapping defined in [6].
e. The transmissions in Step 7 should follow the 4B3B conversion specified in [1], the side-stream scrambler polynomial specified in [2] for master, the generation of Sdn[2:0] specified in [5], and the ternary symbol mapping defined in [8].
f. The transmissions in Step 8 should follow the 4B3B conversion specified in [1], the side-stream scrambler polynomial specified in [2] for master, the generation of Sdn[2:0] specified in [4], and the ternary symbol mapping defined in [9].

Possible Problems: Some devices may not allow configuration as master or slave, in which case only the supported configuration will be tested.

5.2 Test 3.1.2 – PCS Reset

Purpose: To verify that the PCS properly initializes upon receipt of a reset request from the management entity.

References:

[1] IEEE Std. 802.3bw subclause 96.3.1 – PCS Reset function

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
Reference [1] states that the PCS is reset upon power on or the receipt of a reset request from management entity. The PCS Reset function causes the PCS Transmit State Diagram to transition to the SEND IDLE state and the side-stream scrambler to be reset. The initial value of the scrambler shall never be all zeros.

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Procedure:

1. Configure the DUT as master.
2. Monitor and decode the transmissions from the DUT and cause the management to request a PCS Reset.
3. Configure the DUT as slave and repeat step 2.

Observable Results:

a. The DUT should transition to the SEND IDLE state upon reception of a PCS Reset request when configured as master.
b. The DUT should transition to the SEND IDLE state upon reception of a PCS Reset request when configured as slave.
c. The DUT should reset the scrambler to a non-zero value upon reception of a PCS Reset request when configured as master.
d. The DUT should reset the scrambler to a non-zero value upon reception of a PCS Reset request when configured as slave.

Possible Problems: If the ability to control the PCS Reset request is not available, this test cannot be performed. Also, some devices may not allow configuration as master or slave, in which case only the supported configuration will be tested.

5.3 Test 3.1.3 – Transmit Proper SSD

Purpose: To verify that the PCS properly transmits the SSD upon assertion of tx_enable.

References:

[1] IEEE Std. 802.3bw subclause 96.3.3 – PCS transmit
[2] IEEE Std. 802.3bw subclause 96.3.3.3.5 – Generation of ternary symbol (TA_n, TB_n)

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
Reference [1] states that the PCS transmits 6 consecutive systems of SSD to the PMA upon assertion of tx_enable. Reference [2] states that this translates to ternary symbols of (0,0), (0,0), (0,0).

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Procedure:

1. Monitor and decode the transmissions from the DUT as the DUT is sending frames.

Observable Results:

a. The DUT should transmit (0,0), (0,0), (0,0) at the beginning of every frame.

Possible Problems: None.

5.4 Test 3.1.4 – Transmit Proper ESD

Purpose: To verify that the PCS properly transmits the ESD upon de-assertion of tx_enable in the absence of a transmit error.

References:

[1] IEEE Std. 802.3bw subclause 96.3.3 – PCS transmit
[2] IEEE Std. 802.3bw subclause 96.3.3.3.5 – Generation of ternary symbol (TA_n, TB_n)

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
Reference [1] states that the PCS transmits 6 consecutive systems of ESD to the PMA upon de-assertion of tx_enable in the absence of transmit errors. Reference [2] states that this translates to ternary symbols of (0,0), (0,0), (1,1).

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Procedure:

1. Monitor and decode the transmissions from the DUT as the DUT is sending frames.

Observable Results:

a. The DUT should transmit (0,0), (0,0), (1,1) at the end of every frame that does not have transmit errors.

Possible Problems: None.

5.5 Test 3.1.5 – Transmit ESD with tx_error

Purpose: To verify that the PCS properly transmits the ESD_err upon de-assertion of tx_enable in the presence of a transmit error.

References:

[1] IEEE Std. 802.3bw subclause 96.3.3 – PCS transmit
[2] IEEE Std. 802.3bw subclause 96.3.3.3.5 – Generation of ternary symbol (TA_n, TB_n)

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
Reference [1] states that the PCS transmits 6 consecutive systems of ESD_err to the PMA upon de-assertion of tx_enable in the presence of transmit errors. Reference [2] states that this translates to ternary symbols of (0,0), (0,0), (-1,-1).

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Procedure:

1. Monitor and decode the transmissions from the DUT as the DUT is sending frames while causing the DUT to send transmit errors.

Observable Results:

a. The DUT should transmit (0,0), (0,0), (-1,-1) at the end of every frame that has a transmit error.

Possible Problems: It will not be possible to perform this test if there is no method to force the DUT to transmit a frame with a transmit error. This can be accomplished through an MII test station if there is direct access to the MII.

5.6 Test 3.1.6 – Transmission of Stuff Bits

Purpose: To verify that the PCS inserts stuff bits during the 4B3B conversion for MII data.

References:

[1] IEEE Std. 802.3bw subclause 96.3.3.1.2 – 4B3B conversion for MII data

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
Reference [1] states that the DUT shall append stuff bits at the end of every packet that is not a multiple of 3 bits in length.

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Procedure:

1. Monitor and decode the transmissions from the DUT as the DUT is sending frames of several lengths that are a multiple of 3 (when including 8 bytes for preamble), including but not limited to 64 bytes and 67 bytes.
2. Monitor and decode the transmissions from the DUT as the DUT is sending frames of several lengths that are 1 more than a multiple of 3 (when including 8 bytes for preamble), including but not limited to 66 bytes and 69 bytes.
3. Monitor and decode the transmissions from the DUT as the DUT is sending frames of several lengths that are 2 more than a multiple of 3 (when including 8 bytes for preamble), including but not limited to 65 bytes and 68 bytes.

Observable Results:

a. In step 1 the DUT should not transmit stuff bits.
b. In step 2 the DUT should transmit 2 stuff bits.
c. In step 3 the DUT should transmit 1 stuff bit.

Possible Problems: None.

5.7 Test 3.1.7 – tx_error

Purpose: To verify that the PCS properly sets tx_error based on the value of tx_error_mii and tx_enable_mii.

References:

[1] IEEE Std. 802.3bw subclause 96.3.3 – PCS Transmit
[2] IEEE Std. 802.3bw Figure 96-6: 4B3B MII signal conversion

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A test station capable of controlling the MII signals
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
Reference [1] states that the DUT shall set tx_error=TRUE if tx_error_mii is asserted during the packet period. References [1] and [2] indicate that tx_error=TRUE is held until the DUT enters the ERR ESD1 VECTOR state.

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link while the MII station is sending TXD=0000, TX_EN=FALSE, and TX_ER=FALSE.
2. Instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=TRUE for 1 clock cycle, then set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 5 clock cycles, and then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for at least 2 clock cycles.
3. Monitor the transmissions from the DUT.
4. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 7 clock cycles, then set TXD=0000, TX_EN=TRUE, and TX_ER=TRUE for 1 clock cycle, and then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for at least 3 clock cycles in step 2.
5. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 6 clock cycles, then set TXD=0000, TX_EN=FALSE, and TX_ER=TRUE for at least 2 clock cycles in step 2.
6. Repeat steps 1-3, except instruct the MII station to set TXD=0000, TX_EN=FALSE, and TX_ER=TRUE for 1 clock cycle, then set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 6 clock cycles, and then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for at least 2 clock cycles in step 2.

Observable Results:

a. In step 2, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, and ERR_ESD3.
b. In step 4, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, and ERR_ESD3.
c. In step 5, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, and ESD3.

d. In step 6, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, and ESD3

Possible Problems: None.

6 GROUP 2: PCS Transmit State Diagram

Overview:

The tests defined in this section verify the PCS Transmit State Diagram defined for 100BASE-T1 capable PHYs in section 96.3.3.2 of the IEEE 802.3bw 100BASE-T1 Specification.

6.1 Test 3.2.1 – PCS Transmit State Diagram - SEND IDLE State

Purpose: To verify that the DUT properly behaves while in the SEND IDLE state.

References:

[1] IEEE Std. 802.3bw figure 96-7 – PCS Transmit state diagram

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A test station capable of controlling the MII signals
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Transmit State Diagram. The DUT should remain in the SEND IDLE state until tx_enable=TRUE.

Test Setup: Connect the MII of the Device Under Test (DUT) to the MII test station and the 100BASE-T1 interface of the DUT to the Link Partner via the line tap.

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link while the MII station is sending TXD=0, TX_EN=FALSE, and TX_ER=FALSE.
2. Instruct the MII station to send frame data on TXD while keeping TX_EN=FALSE.
3. Monitor the transmissions from the DUT.
4. Repeat steps 1-3, except instruct the MII station to set TX_ER=TRUE in step 2.
5. Repeat steps 1-3, except instruct the MII station to set TX_ER=FALSE and TX_EN=TRUE in step 2.
6. Repeat steps 1-3, except instruct the MII station to set TX_ER=TRUE and TX_EN=TRUE in step 2.

Observable Results:

a. In step 2, the DUT should remain in the SEND IDLE state.
b. In step 4, the DUT should remain in the SEND IDLE state.
c. In step 5, the DUT should transition to the SSD1 VECTOR state and transmit SSD1.
d. In step 6, the DUT should transition to the SSD1 VECTOR state and transmit SSD1.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

6.2 Test 3.2.2 – PCS Transmit State Diagram - SSD1 VECTOR and SSD2 VECTOR states

Purpose: To verify that the DUT properly behaves while in the SSD1 VECTOR and SSD2 VECTOR states.

References:

[1] IEEE Std. 802.3bw figure 96-7 – PCS Transmit state diagram

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A test station capable of controlling the MII signals
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Transmit State Diagram. Regardless of the transmissions on the MII, once the DUT enters the SSD1 VECTOR state the DUT should transmit SSD1, immediately transition to the SSD2 VECTOR state, transmit SSD2, and then immediately transition to the SSD3 VECTOR state.

Test Setup: Connect the MII of the Device Under Test (DUT) to the MII test station and the 100BASE-T1 interface of the DUT to the Link Partner via the line tap.

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link while the MII station is sending TXD=0000, TX_EN=FALSE, and TX_ER=FALSE.
2. Instruct the MII station to set TX_EN=TRUE and TX_ER=FALSE for at least 3 clock cycles.
3. Monitor the transmissions from the DUT.
4. Repeat steps 1-3, except instruct the MII station to set TX_EN=TRUE and TX_ER=TRUE for at least 3 clock cycles in step 2.
5. Repeat steps 1-3, except instruct the MII station to set TX_EN=TRUE and TX_ER=FALSE for 1 clock cycle, and then set TX_EN=TRUE and TX_ER=TRUE for at least 2 clock cycles in step 2.
6. Repeat steps 1-3, except instruct the MII station to set TX_EN=TRUE and TX_ER=FALSE for 1 clock cycle, then set TX_EN=FALSE and TX_ER=FALSE in step 2.
7. Repeat steps 1-3, instruct the MII station to set TX_EN=TRUE and TX_ER=FALSE for 1 clock cycle, then set TX_EN=FALSE and TX_ER=TRUE in step 2.

Observable Results:

a. In step 2, the DUT should transmit SSD1, SSD2, and SSD3.
b. In step 4, the DUT should transmit SSD1, SSD2, and SSD3.
c. In step 5, the DUT should transmit SSD1, SSD2, and SSD3.
d. In step 6, the DUT should transmit SSD1, SSD2, and SSD3.
e. In step 7, the DUT should transmit SSD1, SSD2, and SSD3.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

6.3 Test 3.2.3 – PCS Transmit State Diagram - SSD3 VECTOR state

Purpose: To verify that the DUT properly behaves while in the SSD3 VECTOR state.

References:

[1] IEEE Std. 802.3bw figure 96-7 – PCS Transmit state diagram

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A test station capable of controlling the MII signals
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Transmit State Diagram. Once the DUT enters the SSD3 VECTOR state, the DUT should transmit SSD3 and then transition to TRANSMIT DATA, ERR ESD1 VECTOR, or ESD1 VECTOR.

Test Setup: Connect the MII of the Device Under Test (DUT) to the MII test station and the 100BASE-T1 interface of the DUT to the Link Partner via the line tap.

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link while the MII station is sending TXD=0000, TX_EN=FALSE, and TX_ER=FALSE.
2. Instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for at least 3 clock cycles.
3. Monitor the transmissions from the DUT.
4. Repeat steps 1-3, except instruct the MII station to set TXD=0000, TX_EN=TRUE and TX_ER=FALSE for at least 3 clock cycles in step 2.
5. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 2 clock cycles, and then set TX_EN=TRUE and TX_ER=TRUE for at least 1 clock cycle in step 2.
6. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 2 clock cycles, then set TX_EN=FALSE and TX_ER=FALSE for at least 4 clock cycles in step 2.
7. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=TRUE for 2 clock cycles, then set TX_EN=FALSE and TX_ER=FALSE for at least 4 clock cycles in step 2.

Observable Results:

a. In step 2, the DUT should transmit SSD1 and SSD2, enter the SSD3 VECTOR state and transmit SSD3, and then enter the TRANSMIT DATA state and transmit the scrambled version of "010".

b. In step 4, the DUT should transmit SSD1 and SSD2, enter the SSD3 VECTOR state and transmit SSD3, and then enter the TRANSMIT DATA state and transmit the scrambled version of "000".

c. In step 5, the DUT should transmit SSD1 and SSD2, enter the SSD3 VECTOR state and transmit SSD3, and then enter the TRANSMIT DATA state and transmit the scrambled version of "010".

d. In step 6, the DUT should transmit SSD1 and SSD2, enter the SSD3 VECTOR state and transmit SSD3, enter the ESD1 VECTOR state and transmit ESD1, and then transmit ESD2 and ESD3.

e. In step 7, the DUT should transmit SSD1 and SSD2, enter the SSD3 VECTOR state and transmit SSD3, enter the ERR ESD1 VECTOR state and transmit ESD1, and then transmit ESD2 and ERR_ESD3.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

6.4 Test 3.2.4 – PCS Transmit State Diagram - TRANSMIT DATA state

Purpose: To verify that the DUT properly behaves while in the TRANSMIT DATA state.

References:

[1] IEEE Std. 802.3bw figure 96-7 – PCS Transmit state diagram

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A test station capable of controlling the MII signals
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Transmit State Diagram. Once the DUT enters the TRANSMIT DATA state, it should remain in this state and encode tx_data until TX_EN=FALSE.

Test Setup: Connect the MII of the Device Under Test (DUT) to the MII test station and the 100BASE-T1 interface of the DUT to the Link Partner via the line tap.

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link while the MII station is sending TXD=0000, TX_EN=FALSE, and TX_ER=FALSE.
2. Instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for at least 4 clock cycles.
3. Monitor the transmissions from the DUT.
4. Repeat steps 1-3, except instruct the MII station to set TXD=0000, TX_EN=TRUE and TX_ER=FALSE for at least 4 clock cycles in step 2.
5. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=TRUE for at least 4 clock cycles in step 2.
6. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 4 clock cycles, then set TX_EN=FALSE and TX_ER=FALSE for at least 3 clock cycles in step 2.
7. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=TRUE for 4 clock cycles, then set TX_EN=FALSE and TX_ER=FALSE for at least 3 clock cycles in step 2.

Observable Results:

a. In step 2, the DUT should transmit SSD1, SSD2, and SSD3, then enter the TRANSMIT DATA state and transmit the scrambled version of "010" and "101".
b. In step 4, the DUT should transmit SSD1, SSD2, and SSD3, then enter the TRANSMIT DATA state and transmit the scrambled version of "000" and "000".
c. In step 5, the DUT should transmit SSD1, SSD2, and SSD3, then enter the TRANSMIT DATA state and transmit the scrambled version of "010" and "101".
d. In step 6, the DUT should transmit SSD1, SSD2, and SSD3, then enter the TRANSMIT DATA state and transmit the scrambled version of "010", "101", “0xx”, and then transmit ESD1, ESD2, and ESD3. The transmission of “0xx” represents a data bit of 0 and two stuff bits.

e. In step 7, the DUT should transmit SSD1, SSD2, and SSD3, then enter the TRANSMIT DATA state and transmit the scrambled version of "010", "101", “0xx”, and then transmit ESD1, ESD2, and ERR_ESD3. The transmission of “0xx” represents a data bit of 0 and two stuff bits.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

6.5 Test 3.2.5 – PCS Transmit State Diagram - ESD1 VECTOR state

Purpose: To verify that the DUT properly behaves while in the ESD1 VECTOR state.

References:

[1] IEEE Std. 802.3bw figure 96-7 – PCS Transmit state diagram

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A test station capable of controlling the MII signals
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Transmit State Diagram. Regardless of the transmissions on the MII, once the DUT enters the ESD1 VECTOR state the DUT should transmit ESD1, immediately transition to the ESD2 VECTOR state, and transmit ESD2.

Test Setup: Connect the MII of the Device Under Test (DUT) to the MII test station and the 100BASE-T1 interface of the DUT to the Link Partner via the line tap.

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link while the MII station is sending TXD=0000, TX_EN=FALSE, and TX_ER=FALSE.
2. Instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 6 clock cycles, then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for at least 2 clock cycles.
3. Monitor the transmissions from the DUT.
4. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 5 clock cycles, then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for 1 clock cycle, and then set TX_EN=TRUE in step 2.

Observable Results:

a. In step 2, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1 followed by ESD2.
b. In step 4, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1 followed by ESD2.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

6.6 Test 3.2.6 – PCS Transmit State Diagram - ESD2 VECTOR state

Purpose: To verify that the DUT properly behaves while in the ESD2 VECTOR state.

References:

[1] IEEE Std. 802.3bw figure 96-7 – PCS Transmit state diagram

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A test station capable of controlling the MII signals
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Transmit State Diagram. Regardless of the transmissions on the MII, once the DUT enters the ESD2 VECTOR state the DUT should transmit ESD2, immediately transition to the ESD3 VECTOR state, and transmit ESD3.

Test Setup: Connect the MII of the Device Under Test (DUT) to the MII test station and the 100BASE-T1 interface of the DUT to the Link Partner via the line tap.

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link while the MII station is sending TXD=0000, TX_EN=FALSE, and TX_ER=FALSE.
2. Instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 5 clock cycles, then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for at least 2 clock cycles.
3. Monitor the transmissions from the DUT.
4. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 6 clock cycles, then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for 1 clock cycle, and then set TX_EN=TRUE in step 2.

Observable Results:

a. In step 2, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, and ESD3.

b. In step 4, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, and ESD3.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

6.7 Test 3.2.7 – PCS Transmit State Diagram - ESD3 VECTOR state

Purpose: To verify that the DUT properly behaves while in the ESD3 VECTOR state.

References:

[1] IEEE Std. 802.3bw figure 96-7 – PCS Transmit state diagram

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A test station capable of controlling the MII signals
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Transmit State Diagram. Regardless of the transmissions on the MII, once the DUT enters the ESD3 VECTOR state the DUT should transmit ESD3, immediately transition to the SEND IDLE state, and transmit IDLE.

Test Setup: Connect the MII of the Device Under Test (DUT) to the MII test station and the 100BASE-T1 interface of the DUT to the Link Partner via the line tap.

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link while the MII station is sending TXD=0000, TX_EN=FALSE, and TX_ER=FALSE.
2. Instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=FALSE for 8 clock cycles, then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for at least 3 clock cycles.
3. Monitor the transmissions from the DUT.

Observable Results:

a. In step 2, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, ESD3, and Idle.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

6.8 Test 3.2.8 – PCS Transmit State Diagram - ERR ESD1 VECTOR state

Purpose: To verify that the DUT properly behaves while in the ESD1 VECTOR state.

References:

[1] IEEE Std. 802.3bw figure 96-7 – PCS Transmit state diagram

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A test station capable of controlling the MII signals
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Transmit State Diagram. Regardless of the transmissions on the MII, once the DUT enters the ERR ESD1 VECTOR state the DUT should transmit ESD1, immediately transition to the ERR ESD2 VECTOR state, and transmit ESD2.

Test Setup: Connect the MII of the Device Under Test (DUT) to the MII test station and the 100BASE-T1 interface of the DUT to the Link Partner via the line tap.

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link while the MII station is sending TXD=0000, TX_EN=FALSE, and TX_ER=FALSE.
2. Instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=TRUE for 6 clock cycles, then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for at least 2 clock cycles.
3. Monitor the transmissions from the DUT.
4. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=TRUE for 5 clock cycles, then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for 1 clock cycle, and then set TX_EN=TRUE in step 2.

Observable Results:

a. In step 2, after exiting TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, and ERR_ESD3.
b. In step 4, after exiting TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, and ERR_ESD3.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

6.9 Test 3.2.9 – PCS Transmit State Diagram - ERR ESD2 VECTOR state

Purpose: To verify that the DUT properly behaves while in the ERR ESD2 VECTOR state.

References:

[1] IEEE Std. 802.3bw figure 96-7 – PCS Transmit state diagram

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A test station capable of controlling the MII signals
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Transmit State Diagram. Regardless of the transmissions on the MII, once the DUT enters the ERR ESD2 VECTOR state the DUT should transmit ESD2, immediately transition to the ERR ESD3 VECTOR state, and transmit ERR_ESD3.

Test Setup: Connect the MII of the Device Under Test (DUT) to the MII test station and the 100BASE-T1 interface of the DUT to the Link Partner via the line tap.

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link while the MII station is sending TXD=0000, TX_EN=FALSE, and TX_ER=FALSE.
2. Instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=TRUE for 5 clock cycles, then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for at least 2 clock cycles.
3. Monitor the transmissions from the DUT.
4. Repeat steps 1-3, except instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=TRUE for 6 clock cycles, then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for 1 clock cycle, and then set TX_EN=TRUE in step 2.

Observable Results:

a. In step 2, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, and ERR_ESD3.
b. In step 4, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, and ERR_ESD3.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

6.10 Test 3.2.10 – PCS Transmit State Diagram - ERR ESD3 VECTOR state

Purpose: To verify that the DUT properly behaves while in the ERR ESD3 VECTOR state.

References:

[1] IEEE Std. 802.3bw figure 96-7 – PCS Transmit state diagram

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A test station capable of controlling the MII signals
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Transmit State Diagram. Regardless of the transmissions on the MII, once the DUT enters the ERR ESD3 VECTOR state the DUT should transmit ERR_ESD3, immediately transition to the SEND IDLE state, and transmit IDLE.

Test Setup: Connect the MII of the Device Under Test (DUT) to the MII test station and the 100BASE-T1 interface of the DUT to the Link Partner via the line tap.

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link while the MII station is sending TXD=0000, TX_EN=FALSE, and TX_ER=FALSE.
2. Instruct the MII station to set TXD=0101, TX_EN=TRUE, and TX_ER=TRUE for 8 clock cycles, then set TXD=0000, TX_EN=FALSE, and TX_ER=FALSE for at least 3 clock cycles.
3. Monitor the transmissions from the DUT.

Observable Results:

a. In step 2, after exiting the TRANSMIT DATA state, the DUT should transmit ESD1, ESD2, ERR_ESD3, and Idle.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

7 GROUP 3: PCS Receive

Overview:

The tests defined in this section verify the PCS Receive process defined for 100BASE-T1 capable PHYs in section 96.3.4 of the IEEE 802.3bw 100BASE-T1 Specification.

7.1 Test 3.3.1 – Receive PCS Signaling

Purpose: To verify that the PCS properly decodes 100BASE-T1 signaling.

References:

[1] IEEE Std. 802.3bw subclause 96.3.4 – PCS Receive

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:

The 100BASE-T1 PCS uses a combination of a 4B3B conversion for MII data, side-stream scrambling, and ternary symbols to encode the data for transmission on the line. In addition to the scrambler, there are additional functions performed on the data and idle eliminate the correlation transmit data and to balance the power density.

This test is designed to ensure that the DUT properly decodes the 100BASE-T1 signaling during training, idle transmission, and data transmission.

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Test Procedure:

1. Configure the DUT as MASTER and the Link Partner as SLAVE.
2. Connect the DUT to a Link Partner and configure both devices to perform training and establish a link. Monitor the management indications from the DUT.
3. Connect the DUT to a Link Partner, establish a link, and configure the Link Partner to transmit valid frames to the DUT. Monitor the management indications from the DUT.
4. Connect the DUT to a Link Partner, establish a link, and configure the Link Partner to transmit Idle without frames. Monitor the management indications from the DUT.
5. Configure the DUT as SLAVE and the Link Partner as MASTER.
6. Connect the DUT to a Link Partner and configure both devices to perform training and establish a link. Monitor the management indications from the DUT.
7. Connect the DUT to a Link Partner, establish a link, and configure the Link Partner to transmit valid frames to the DUT. Monitor the management indications from the DUT.
8. Connect the DUT to a Link Partner, establish a link, and configure the Link Partner to transmit Idle without frames. Monitor the management indications from the DUT.

Observable Results:

a. In step 2 The DUT should accept the idle pattern and establish a link.
b. In step 3 The DUT should indicate link and should receive valid frames.
c. In step 4 the DUT should indicate reception of idle and maintain the link.
d. In step 6 The DUT should accept the idle pattern and establish a link.
e. In step 7 The DUT should indicate link and should receive valid frames.
f. In step 8 the DUT should indicate reception of idle and maintain the link.

Possible Problems: None.

7.2 Test 3.3.2 – Automatic Polarity Detection (Optional)

Purpose: To verify that the PCS properly detects polarity.

References:

[1] IEEE Std. 802.3bw subclause 96.3.4.4 – PCS Receive automatic polarity detection (optional)

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
Reference [1] states that the DUT shall automatically detect the polarity of the received signal.

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Procedure:

1. Connect the DUT to the Link Partner, transmit several frames and observe that the link has successfully been established.
2. Switch the polarity by reversing the pair connecting the DUT to the Link Partner. Transmit several frames and observe that the link has successfully been established.

Observable Results:

a. The DUT should establish a link regardless of the receive polarity.

Possible Problems: Automatic Polarity Detection is an optional function. This test cannot be completed if the DUT does not implement Automatic Polarity Detection.

7.3 Test 3.3.3 – Receive SSD

Purpose: To verify that the PCS properly accepts frames with valid SSD.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram
[1] IEEE Std. 802.3bw subclause 96.3.4.2 – PCS Receive symbol decoding

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
The DUT should accept frames with valid SSD of (0,0), (0,0), (0,0).

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Procedure:

1. Transmit frames with valid SSD to the DUT while monitoring the DUT transmissions and counters.

Observable Results:

a. The DUT should accept all frames.

Possible Problems: None.

7.4 Test 3.3.4 – Receive ESD

Purpose: To verify that the PCS properly accepts a frame with valid ESD.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram
[1] IEEE Std. 802.3bw subclause 96.3.4.2 – PCS Receive symbol decoding

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
The DUT should accept the frame with valid ESD of (0,0), (0,0), (1,1).

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Procedure:

1. Transmit valid frames to the DUT while monitoring the DUT transmissions and counters.

Observable Results:

a. The DUT should accept all frames.

Possible Problems: None.

7.5 Test 3.3.5 – Receive ERR_ESD3

Purpose: To verify that the PCS properly indicates reception of an error upon reception of ERR_ESD3.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram
[1] IEEE Std. 802.3bw subclause 96.3.4.2 – PCS Receive symbol decoding

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
Reference [1] shows that the DUT transitions to the RX_ERROR state and [2] states that the DUT shall set pcs_rx_er=TRUE upon reception of ERR_ESD3.

Test Setup: Connect the Device Under Test (DUT) to the test station via the line tap.

Procedure:

1. Transmit a frame with ERR_ESD3 to the DUT while monitoring the DUT transmissions and counters.

Observable Results:

a. The DUT should indicate reception of an errored frame.

Possible Problems: None.

7.6 Test 3.3.6 – Reception of Stuff Bits

Purpose: To verify that the PCS removes stuff bits during the 3B4B conversion for MII data.

References:

[1] IEEE Std. 802.3bw subclause 96.3.4.2 – PCS Receive symbol decoding

Resource Requirements:

- A test station capable of capturing and decoding ternary symbols
- A compliant 100BASE-T1 Link Partner
- A line tap (Refer to appendix 3.C)

Discussion:
Reference [1] states that the DUT shall discard stuff bits that are inserted during the 4B3B transmit process.

Test Setup: Connect the Device Under Test (DUT) to the Link Partner via the line tap.

Procedure:

1. Transmit frames of several lengths that are a multiple of 3 (when including 8 bytes for preamble), including but not limited to 64 bytes and 67 bytes, to the DUT while monitoring the DUT transmissions and counters.
2. Transmit frames of several lengths that are 1 more than a multiple of 3 (when including 8 bytes for preamble), including but not limited to 66 bytes and 69 bytes, to the DUT while monitoring the DUT transmissions and counters.
3. Transmit frames of several lengths that are 2 more than a multiple of 3 (when including 8 bytes for preamble), including but not limited to 65 bytes and 68 bytes, to the DUT while monitoring the DUT transmissions and counters.

Observable Results:

a. The DUT should accept all frames.

Possible Problems: None.

7.7 Test 3.3.7 – De-Interleave Ternary Pairs

Purpose: To verify that the PCS receiver properly De-Interleaves the received ternary symbols in either ordering.

References:

[1] IEEE Std. 802.3bw subclause 96.3.3.3.10 – Generation of symbol sequence

Resource Requirements:

- A test station capable of transmitting ternary symbols in either interleave order

Discussion:
Reference [1] states that the DUT shall de-interleave the serial stream of ternary symbols as (TA_n, TB_n) or (TB_n, TA_n) to match the interleave order of the transmitter.

Test Setup: Connect the Device Under Test (DUT) to the Test Station.

Procedure:

1. Configure the test station to interleave the serial stream of ternary symbols as (TA_n, TB_n).
2. Connect the DUT to the test station, perform training, establish a link, and transmit frames. Monitor the management indications from the DUT.
3. Repeat steps 1-2, but configure the test station to interleave the serial stream of ternary symbols as (TB_n, TA_n).

Observable Results:

a. In step 2, the DUT should establish a valid link and receive the transmitted frames.
b. In step 3, the DUT should establish a valid link and receive the transmitted frames.

Possible Problems: None.

8 GROUP 4: PCS Receive State Diagram

Overview:

The tests defined in this section verify the PCS Receive State Diagram defined for 100BASE-T1 capable PHYs in section 96.3.4 of the IEEE 802.3bw 100BASE-T1 specification.

8.1 Test 3.4.1 – Receive State Diagram - IDLE State

Purpose: To verify that the DUT behaves properly while in the IDLE state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. The DUT should remain in the IDLE state while it is receiving Idle.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send 3 SSD symbols.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 1-3, except instruct the test station to send a Ternary symbol that does not represent SSD1 or a valid Idle code in step 2.

Observable Results:

a. In step 2, the DUT should set RX_DV=TRUE and RXD=0101 for two clock cycles, then RXD=xxx1 where x is determined by the test pattern sent from the testing station.

b. In step 4, the DUT should set RX_ER=TRUE.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

8.2 Test 3.4.2 – Receive State Diagram - CHECK SSD2 State

Purpose: To verify that the DUT behaves properly while in the CHECK SSD2 state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:
Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. Once the DUT is in the CHECK SSD2 state, it should transition to the CHECK SSD3 state if it receives SSD2 or transition to the BAD SSD state if it receives anything else.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send 3 SSD symbols.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send SSD1 followed by a Ternary symbol other than SSD2 in step 2.

Observable Results:

a. In step 2, the DUT should set RX_DV=TRUE and RXD=0101 .
b. In step 4, the DUT should set RX_ER=TRUE.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

8.3 Test 3.4.3 – Receive State Diagram - CHECK SSD3 State

Purpose: To verify that the DUT behaves properly while in the CHECK SSD3 state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:
Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. Once the DUT is in the CHECK SSD3 state, it should transition to the SSD state if it receives SSD3 or transition to the BAD SSD state if it receives anything else.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send 3 SSD symbols.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send 2 SSD symbols followed by a Ternary symbol other than SSD3 in step 2.

Observable Results:

a. In step 2, the DUT should set RX_DV=TRUE and RXD=0101.
b. In step 4, the DUT should set RX_ER=TRUE.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

8.4 Test 3.4.4 – Receive State Diagram - SSD State

Purpose: To verify that the DUT behaves properly while in the SSD state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:
Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. Regardless of what is being received, once the DUT enters the SSD state it should set transition to the FIRST SSD state.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send 3 SSD symbols followed by a Ternary symbol for 010.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols followed by a Ternary symbol for data 000 in step 3.
5. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols followed by the Ternary symbol for SSD1 in step 3.

Observable Results:

a. In step 2, the DUT should transition to FIRST SSD and set RX_DV=TRUE and RXD=0101.
b. In step 4, the DUT should transition to FIRST SSD and set RX_DV=TRUE and RXD=0101.
c. In step 5, the DUT should transition to FIRST SSD and set RX_DV=TRUE and RXD=0101

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

8.5 Test 3.4.5 – Receive State Diagram - BAD SSD State

Purpose: To verify that the DUT behaves properly while in the BAD SSD state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram
[2] IEEE Std. 802.3bw subclause 96.3.4.1.1 – Variables

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:
Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. The DUT enters the BAD SSD state upon reception of an error in the IDLE, CHECK SSD2, or CHECK SSD3 states. While in BAD SSD the DUT sets RX_ER=TRUE. The DUT waits for check_idle=TRUE to transition back to the IDLE state. The DUT should set check_idle=TRUE upon reception of 6 consecutive valid Idle symbols.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send several Ternary symbols that do not represent SSD or valid Idle.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send SSD followed by several Ternary symbols that do not represent SSD or valid Idle in step 2.
5. Repeat steps 2-3, except instruct the test station to send 2 SSD symbols followed by several Ternary symbols that do not represent SSD or valid Idle in step 2.
6. Repeat steps 2-3, except instruct the test station to transmit a Ternary symbol that does not represent SSD or valid Idle, followed by 1 Idle symbol and a valid frame in step 2.
7. Repeat step 6, except send additional Idle symbols before the frame until the DUT is observed to accept the frame.
8. Repeat steps 6 and 7, except send an SSD symbol, 1 Ternary symbol that does not represent SSD or valid Idle, 1 Idle symbol, and a valid frame in step 6. Increase the number of Idle symbols until the DUT is observed to accept the frame in step 7.
9. Repeat steps 6 and 7, except sendg 2 SSD symbols, 1 Ternary symbol that does not represent SSD or valid Idle, 1 Idle symbol, and a valid frame in step 6. Increase the number of Idle symbols until the DUT is observed to accept the frame in step 7.

Observable Results:

a. In step 2, the DUT should set RX_ER=TRUE while in the BAD SSD state.
b. In step 4, the DUT should set RX_ER=TRUE while in the BAD SSD state.
c. In step 5, the DUT should set RX_ER=TRUE while in the BAD SSD state.
d. In step 7, the DUT should accept the frame upon reception of 6 Idle symbols after the non-SSD and non-Idle symbol.

e. In step 8, the DUT should accept the frame upon reception of 6 Idle symbols after the non-SSD and non-Idle symbol.
f. In step 9, the DUT should accept the frame upon reception of 6 Idle symbols after the non-SSD and non-Idle symbol.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

8.6 Test 3.4.6 – Receive State Diagram - FIRST SSD State

Purpose: To verify that the DUT behaves properly while in the FIRST SSD state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:
Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. Regardless of what is being received, once the DUT enters the FIRST SSD state it should set rx_data = 101 and transition to the SECOND SSD state.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send 3 SSD symbols followed by Ternary symbols for 010 and 101.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols followed by the Ternary symbols for 101 and 000 in step 2.
5. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols followed by the Ternary symbols for 010 and SSD1 in step 2.

Observable Results:

a. In step 2, the DUT should set RX_DV=TRUE and RXD=0101.
b. In step 4, the DUT should set RX_DV=TRUE and RXD=0101.
c. In step 5, the DUT should set RX_DV=TRUE and RXD=0101.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

8.7 Test 3.4.7 – Receive State Diagram - SECOND SSD State

Purpose: To verify that the DUT behaves properly while in the SECOND SSD state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:
Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. Regardless of what is being received, once the DUT enters the SECOND SSD state it should set rx_data = 010 and transition to the THIRD SSD state.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send 3 SSD symbols, and then Ternary symbols for 010, 101, and 010.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, and then Ternary symbols for 010, 101, and 000 in step 2.
5. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, and then Ternary symbols for 010, 101, and the Ternary symbol for SSD1 in step 2.

Observable Results:

a. In step 2, the DUT should set RX_DV=TRUE and RXD=0101.
b. In step 4, the DUT should set RX_DV=TRUE and RXD=0101.
c. In step 5, the DUT should set RX_DV=TRUE and RXD=0101.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

8.8 Test 3.4.8 – Receive State Diagram - THIRD SSD State

Purpose: To verify that the DUT behaves properly while in the THIRD SSD state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:
Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. Once the DUT enters the THIRD SSD state it should set rx_data = 101 and transition to the CHECK ESD2 state if it receives ESD, or transition to the DATA state if it receives anything but ESD.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, and several Ternary symbols representing data.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, the Ternary symbols for 010 and 101, 010, and ESD1 in step 2.
5. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols and several Ternary symbols for 000 in step 2.

Observable Results:

a. In step 2, the DUT should set RX_DV=TRUE and RXD=0101 for 6 clock cycles and then transmit data.
b. In step 4, the DUT should set RX_DV=TRUE and RXD=0101 for 4 clock cycles.
c. In step 5, the DUT should set RX_DV=TRUE and RXD=0101 for 2 clock cycles, RXD=0001 for 1 clock cycle, and then several clock cycles of RXD=0000.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

8.9 Test 3.4.9 – Receive State Diagram - DATA State

Purpose: To verify that the DUT behaves properly while in the DATA state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:
Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. Once the DUT enters the DATA state, it should not exit until it receives ESD1.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, and several Ternary symbols representing data, followed by ESD1.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, and ESD1 in step 2.
5. Repeat step 4, sending additional data before the ESD1 in step 2.

Observable Results:

a. In step 2, the DUT should set RX_DV=TRUE and RXD=0101 for 6 clock cycles and then transmit data before setting RX_DV=FALSE and RXD=0000.
b. In step 4, the DUT should set RX_DV=TRUE and RXD=0101 for 4 clock cycles.
c. In step 5, the DUT should set RX_DV=TRUE and RXD=0101 for 4 clock cycles and transmit the data sent before the ESD.

Possible Problems: In step 5 the DUT may not transmit 1 or 2 of the bits sent prior to the ESD because they may be received as stuff bits during the 3B4B conversion. This test cannot be performed if direct access to the DUT MII signals is not available.

8.10 Test 3.4.10 – Receive State Diagram - CHECK ESD2 State

Purpose: To verify that the DUT behaves properly while in the CHECK ESD2 state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. Once the DUT enters the CHECK ESD2 state, it should transition to CHECK ESD3 upon reception of ESD2 or transition to BAD END if it receives anything other than ESD2.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, and several Ternary symbols representing data, followed by ESD1, ESD2, and ESD3.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, ESD1, and the Ternary symbol for 000 in step 2.
5. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, ESD1, and ESD3 in step 2.

Observable Results:

a. In step 2, the DUT should set RX_DV=TRUE and RXD=0101 for 6 clock cycles and then transmit data, before setting RX_DV=FALSE and RXD=0000.

b. In step 4, the DUT should set RX_DV=TRUE and RXD=0101 for 4 clock cycles, and then set RX_ER=TRUE.

c. In step 5, the DUT should set RX_DV=TRUE and RXD=0101 for 4 clock cycles, and then set RX_ER=TRUE.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

8.11 Test 3.4.11 – Receive State Diagram - CHECK ESD3 State

Purpose: To verify that the DUT behaves properly while in the CHECK ESD3 state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. Once the DUT enters the CHECK ESD3 state, it should transition to ESD upon reception of ESD3 or, RX ERROR upon reception of ERR_ESD3, or BAD END if it receives anything else.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, and several Ternary symbols representing data, followed by ESD1, ESD2, and ESD3.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, ESD1, ESD2, and ERR_ESD3 in step 2.
5. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, ESD1, ESD2, and the Ternary symbol for 000 in step 2.
6. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, ESD1, ESD2, ESD2, and ESD3 in step 2.

Observable Results:

a. In step 2, the DUT should set RX_DV=TRUE and RXD=0101 for 6 clock cycles and then transmit data, before setting RX_DV=FALSE and RXD=0000.
b. In step 4, the DUT should set RX_DV=TRUE and RXD=0101 for 5 clock cycles, and then set RX_ER=TRUE.
c. In step 5, the DUT should set RX_DV=TRUE and RXD=0101 for 5 clock cycles, and then set RX_ER=TRUE.
d. In step 6, the DUT should set RX_DV=TRUE and RXD=0101 for 5 clock cycles, and then set RX_ER=TRUE.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

8.12 Test 3.4.12 – Receive State Diagram - BAD ESD2 State

Purpose: To verify that the DUT behaves properly while in the BAD ESD2 state.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. Once the DUT enters the BAD ESD2 state, it should set RX_ER = TRUE and transition to BAD END regardless of what is received.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instructing the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, ESD1, and the Ternary symbol for 000 twice.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, ESD1, the Ternary symbol for 000, and ESD3 in step 2.
5. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, ESD1, the Ternary symbol for 000, and SSD in step 2.

Observable Results:

a. In step 2, the DUT should set RX_DV=TRUE and RXD=0101 for 4 clock cycles, and then set RX_ER=TRUE for 2 clock cycles.
b. In step 4, the DUT should set RX_DV=TRUE and RXD=0101 for 4 clock cycles, and then set RX_ER=TRUE for 2 clock cycles.
c. In step 5, the DUT should set RX_DV=TRUE and RXD=0101 for 4 clock cycles, and then set RX_ER=TRUE for 2 clock cycles.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

8.13 Test 3.4.13 – Receive State Diagram - BAD END and RX ERROR states

Purpose: To verify that the DUT behaves properly while in the BAD END or RX ERROR states.

References:

[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:

Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the PCS Receive State Diagram. Once the DUT enters the BAD END or RX ERROR state, it should set RX_ER=TRUE and transition to IDLE regardless of what is received.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, ESD1, ESD2, and ERR ESD3.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, ESD1, ESD2, and all ternary symbols that do not represent ESD3 or ERR_ESD3 in step 2.
5. Repeat steps 2-3, except instruct the test station to send 3 SSD symbols, the Ternary symbols for 010, 101, 010, 101, 010, ESD1, ESD2, and SSD in step 2.

Observable Results:

a. In step 2, the DUT should set RX_DV=TRUE and RXD=0101 for 5 clock cycles, and then set RX_ER=TRUE for 1 clock cycle.
b. In step 4, the DUT should set RX_DV=TRUE and RXD=0101 for 5 clock cycles, and then set RX_ER=TRUE for 1 clock cycle.
c. In step 5, the DUT should set RX_DV=TRUE and RXD=0101 for 5 clock cycles, and then set RX_ER=TRUE for 1 clock cycle.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

9 GROUP 5: JAB State Diagram

Overview:

The tests defined in this section verify the JAB State Diagram defined for 100BASE-T1 capable PHYs in section 96.3.4 of the IEEE 802.3bw 100BASE-T1 specification.

9.1 Test 3.5.1 – JAB State Diagram - rcv_max_timer

Purpose: To verify that the DUT properly implements a rcv_max_timer of 1.08 ms +/- 54 µs.

References:

[1] IEEE Std. 802.3bw figure 96-11 – JAB state Diagram
[1] IEEE Std. 802.3bw figure 96-10 – PCS Receive state diagram
[2] IEEE Std. 802.3bw subclause 96.3.4.1.3 – Timer

Resource Requirements:

- A test station capable of transmitting ternary symbols
- A test station capable of capturing the MII signals from the DUT

Discussion:
Reference [1] shows the valid transitions that a 100BASE-T1 device can take through the JAB State Diagram. Once the DUT sets receiving=TRUE it enters the MONJAB state and starts the rcv_max_timer. Upon expiration of the timer, the DUT transitions to the JAB state, which causes the PCS Receive State Diagram to transition to the IDLE state. The value of rcv_max_timer should be within the range of 1.08 ms +/- 54 µs.

This test will be performed by sending a very large packet that is terminated with ESD1, ESD2, and ERR_ESD3. If the frame is not long enough to cause a transition to the JAB state then the PCS Receive State Diagram will transition to the RX ERROR state and set pcs_rx_er =TRUE for one clock cycle while pcs_rx_dv =TRUE. If the frame is long enough to cause a transition to the JAB state then the PCS Receive State Diagram will transition directly to the IDLE state where it sets pcs_rx_dv=FALSE. The remaining frame data and/or ESD symbols will cause a transition to the BAD SSD state where the DUT will set pcs_rx_er=TRUE while pcs_rx_dv=FALSE.

Test Setup: Connect the Device Under Test (DUT) to the test stations via the 100BASE-T1 and MII interfaces.

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send Preamble and frame data for 1.026 ms, followed by ESD1, ESD2, and ERR_ESD3.
3. Monitor the MII transmissions from the DUT.
4. Repeat steps 2-3, increasing the amount of data until the PCS Receive State Diagram transitions directly to the IDLE state.

Observable Results:

a. The DUT should implement a rcv_max_timer within the range of 1.08 ms +/- 54 µs.

Possible Problems: This test cannot be performed if direct access to the DUT MII signals is not available.

10 TEST SUITE APPENDICES

Overview:
The appendices contained in this section are intended to provide additional low-level technical details pertinent to specific tests defined in this test suite. Test suite appendices often cover topics that are beyond the scope of the standard, but are specific to the methodologies used for performing the measurements covered in this test suite. This may also include details regarding a specific interpretation of the standard (for the purposes of this test suite), in cases where a specification may appear unclear or otherwise open to multiple interpretations.

Scope:
Test suite appendices are considered informative, and pertain only to tests contained in this test suite.

10.1 Appendix 3.A –Testing Devices Without MII Access

Purpose: To detail the tests that can be performed on a device without MII access, and to detail the modifications to the Procedures and Observable Results.

Discussion: The amount of tests that can be performed on a device without MII access is reduced. Typically, the only observation points that will be available packet counters or packets that are forwarded through another port on the DUT. Also, the only transmissions available from the DUT will be Training, Idle, and packets. The following table lists all tests and if they are possible with or without modification. The necessary modifications are detailed in the following sections.

Test Name	Test Number	No Modification Required	Requires Modification	Cannot be Tested
Group 1: PCS Transmit				
PCS Signaling	Test 3.1.1	All Parts		
PCS Reset	Test 3.1.2	All Parts		
Transmit Proper SSD	Test 3.1.3	All Parts		
Transmit Proper ESD	Test 3.1.4	All Parts		
Transmit ESD with tx_error	Test 3.1.5			All Parts
Transmission of Stuff Bits	Test 3.1.6	All Parts		
Tx_error	Test 3.1.7			All Parts
Group 2: PCS Transmit State Diagram				
Transmit State Diagram - SEND IDLE state	Tests 3.2.1		c	a, b, d
Transmit State Diagram - SSD1 VECTOR and SSD2 VECTOR states	Tests 3.2.2		a	b, c, d, e
Transmit State Diagram - SSD3 VECTOR state	Tests 3.2.3		a	b, c, d, e
Transmit State Diagram - TRANSMIT DATA state	Tests 3.2.4		a	b, c, d, e
Transmit State Diagram - ESD1 VECTOR state	Tests 3.2.5		a	b
Transmit State Diagram - ESD2 VECTOR state	Tests 3.2.6		a	b
Transmit State Diagram - ESD3 VECTOR state	Tests 3.2.7		a	
Transmit State Diagram - ERR ESD1 VECTOR state	Tests 3.2.8			All Parts
Transmit State Diagram - ERR ESD2 VECTOR state	Tests 3.2.9			All Parts
Transmit State Diagram - ERR ESD2 VECTOR state	Tests 3.2.10			All Parts
Group 3: PCS Receive				
Receive PCS Signaling	Test 3.3.1	All Parts		
Automatic Polarity Detection (Optional)	Test 3.3.2	All Parts		
Receive SSD	Test 3.3.3	All Parts		
Receive ESD	Test 3.3.4	All Parts		
Receive ERR_ESD3	Test 3.3.5	All Parts		
Reception of Stuff Bits	Test 3.3.6	All Parts		
De-Interleave Ternary Pairs	Test 3.3.7	All Parts		
Group 4: PCS Receive State Diagram				
Receive State Diagram - IDLE State	Test 3.4.1		a, b	
Receive State Diagram - CHECK SSD2 State	Test 3.4.2		a, b	
Receive State Diagram - CHECK SSD3 State	Test 3.4.3		a, b	
Receive State Diagram –SSD State	Test 3.4.4		a	b, c
Receive State Diagram - BAD SSD State	Test 3.4.5		d, e, f	a, b, c
Receive State Diagram - FIRST SSD State	Test 3.4.6		a	b, c
Receive State Diagram - SECOND SSD State	Test 3.4.7		a	b, c
Receive State Diagram - THIRD SSD State	Test 3.4.8		a	b, c
Receive State Diagram - DATA State	Test 3.4.9		a	b, c
Receive State Diagram - CHECK ESD2 State	Test 3.4.10		All Parts	
Receive State Diagram - CHECK ESD3 State	Test 3.4.11		All Parts	
Receive State Diagram - BAD ESD2 State	Test 3.4.12		All Parts	
Receive State Diagram - BAD END and RX ERROR States	Test 3.4.13		All Parts	
Group 5: JAB State Diagram				
JAB State Diagram - rcv_max_timer	Test 3.5.1			All Parts

Table 3A. 1 - Tests that are possible without MII access

10.1.1 Test 3.2.1 – PCS Transmit State Diagram - SEND IDLE State

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link.
2. Instruct the DUT to transmit a valid packet.
3. Monitor the transmissions from the DUT.

Observable Results:

c. In step 2, the DUT should transmit valid SSD1.

10.1.2 Test 3.2.2 – PCS Transmit State Diagram - SSD1 VECTOR and SSD2 VECTOR states

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link.
2. Instruct the DUT to transmit a valid packet.
3. Monitor the transmissions from the DUT.

Observable Results:

a. In step 2, the DUT should transmit valid SSD1, SSD2, and SSD3.

10.1.3 Test 3.2.3 – PCS Transmit State Diagram - SSD3 VECTOR state

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link.
2. Instruct the DUT to transmit a valid packet.
3. Monitor the transmissions from the DUT.

Observable Results:

a. In step 2, the DUT should transmit valid SSD3 and frame data.

10.1.4 Test 3.2.4 – PCS Transmit State Diagram - TRANSMIT DATA state

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link.
2. Instruct the DUT to transmit a valid packet.
3. Monitor the transmissions from the DUT.

Observable Results:

a. In step 2, the DUT should transmit valid frame data and ESD.

10.1.5 Test 3.2.5 – PCS Transmit State Diagram - ESD1 VECTOR state

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link.
2. Instruct the DUT to transmit a valid packet.
3. Monitor the transmissions from the DUT.

Observable Results:

a. In step 2, the DUT should transmit valid ESD1 and ESD2.

10.1.6 Test 3.2.6 – PCS Transmit State Diagram - ESD2 VECTOR state

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link.
2. Instruct the DUT to transmit a valid packet.
3. Monitor the transmissions from the DUT.

Observable Results:

a. In step 2, the DUT should transmit valid ESD2 and ESD3.

10.1.7 Test 3.2.7 – PCS Transmit State Diagram - ESD3 VECTOR state

Procedure:

1. Configure the test station and Link Partner such that the DUT establishes a valid link.
2. Instruct the DUT to transmit a valid packet.
3. Monitor the transmissions from the DUT.

Observable Results:

a. In step 2, the DUT should transmit valid ESD3 followed by Idle.

10.1.8 Test 3.4.1 – Receive State Diagram - IDLE State

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send a valid packet.
3. Monitor the DUT to determine if the DUT accepted the packet.
4. Repeat steps 1-3, except instruct the test station to send a Ternary symbol that does not represent SSD1 or valid Idle, followed by 1 Idle symbol and a valid packet in step 2.

Observable Results:

a. In step 2, the DUT should receive the packet.
b. In step 4, the DUT should not receive the packet.

10.1.9 Test 3.4.2 – Receive State Diagram - CHECK SSD2 State

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send a valid packet.
3. Monitor the DUT to determine if the DUT accepted the packet.
4. Repeat steps 1-3, except instruct the test station to send SSD1 and a Ternary symbol that does not represent SSD2, followed by 1 Idle symbol and a valid packet in step 2.
5. Repeat steps 1-3, except instruct the test station to transmit a packet where SSD2 of an otherwise valid packet is replaced with a Ternary symbol that does not represent SSD2.

Observable Results:

a. In step 2, the DUT should receive the packet.
b. In steps 4 and 5, the DUT should not receive the packet.

10.1.10 Test 3.4.3 – Receive State Diagram - CHECK SSD3 State

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send a valid packet.
3. Monitor the DUT to determine if the DUT accepted the packet.
4. Repeat steps 1-3, except instruct the test station to send SSD1, SSD2, and a Ternary symbol that does not represent SSD3, followed by 1 Idle symbol and a valid packet in step 2.
5. Repeat steps 1-3,except instruct the test station to transmit a packet where SSD3 of an otherwise valid packet is replaced with a Ternary symbol that does not represent SSD3 in step 2.

Observable Results:

a. In step 2, the DUT should receive the packet.
b. In steps 4 and 5, the DUT should not receive the packet.

10.1.11 Test 3.4.4 – Receive State Diagram - SSD State

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send a valid packet.
3. Monitor the DUT to determine if the DUT accepted the packet.

Observable Results:

a. In step 2, the DUT should receive the packet.

10.1.12 Test 3.4.5 – Receive State Diagram - BAD SSD State

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send a Ternary symbols that does not represent SSD or valid Idle, followed by 1 Idle symbol and a valid packet.
3. Monitor the DUT to determine if the DUT accepted the packet.
4. Repeat steps 2-3, sending additional Idle before the packet until the DUT is observed to accept the frame.
5. Repeat steps 1-4, except instruct the test station to send 1 SSD symbol followed by 1 Ternary symbols that does not represent SSD or valid Idle, followed by 1 Idle symbol and a valid packet in step 2. Increase the number of Idle symbols until the DUT is observed to accept the frame.
6. Repeat steps 1-4, except instruct the test station to send 2 SSD symbols followed by 1 Ternary symbols that does not represent SSD or valid Idle, followed by 1 Idle symbol and a valid packet in step 2. Increase the number of Idle symbols until the DUT is observed to accept the frame.

Observable Results:

d. In step 4, the DUT should accept the frame upon reception of 6 Idle symbols after the non-SSD and non-Idle symbol.
e. In step 5, the DUT should accept the frame upon reception of 6 Idle symbols after the non-SSD and non-Idle symbol.
f. In step 6, the DUT should accept the frame upon reception of 6 Idle symbols after the non-SSD and non-Idle symbol.

10.1.13 Test 3.4.6 – Receive State Diagram - FIRST SSD State

Procedure:

4. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
5. Instruct the test station to send a valid packet.
6. Monitor the DUT to determine if the DUT accepted the packet.

Observable Results:

b. In step 2, the DUT should receive the packet.

10.1.14 Test 3.4.7 – Receive State Diagram - SECOND SSD State

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send a valid packet.
3. Monitor the DUT to determine if the DUT accepted the packet.

Observable Results:

a. In step 2, the DUT should receive the packet.

10.1.15 Test 3.4.8 – Receive State Diagram - THIRD SSD State

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send a valid packet.
3. Monitor the DUT to determine if the DUT accepted the packet.

Observable Results:

a. In step 2, the DUT should receive the packet.

10.1.16 Test 3.4.9 – Receive State Diagram - DATA State

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send a valid packet.
3. Monitor the DUT to determine if the DUT accepted the packet.

Observable Results:

a. In step 2, the DUT should receive the packet.

10.1.17 Test 3.4.10 – Receive State Diagram - CHECK ESD2 State

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send a valid packet.
3. Monitor the DUT to determine if the DUT accepted the packet.
4. Repeat steps 1-3, except instruct the test station to send preamble and valid frame data followed by ESD1, the Ternary symbol for 000, ESD2, and ESD3 in step 2.
5. Repeat steps 1-3, except instruct the test station to send preamble and valid frame data followed by ESD1, ESD3, ESD2, and ESD3 in step 2.

Observable Results:

a. In step 2, the DUT should receive the packet.
b. In step 4, the DUT should indicate the reception of an errored packet.
c. In step 5, the DUT should indicate the reception of an errored packet.

10.1.18 Test 3.4.11– Receive State Diagram - CHECK ESD3 State

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send a valid packet.
3. Monitor the DUT to determine if the DUT accepted the packet.
4. Repeat steps 1-3, except instruct the test station to send preamble and valid frame data followed by ESD1, ESD2, and ERR_ESD3 in step 2.
5. Repeat steps 1-3, except instruct the test station to send preamble and valid frame data followed by ESD1, ESD2, the Ternary symbol for 000, and ESD3 in step 2.
6. Repeat steps 1-3, except instruct the test station to send preamble and valid frame data followed by ESD1, ESD2, ESD2, and ESD3 in step 2.

Observable Results:

a. In step 2, the DUT should receive the packet.
b. In step 4, the DUT should indicate the reception of an errored packet.
c. In step 5, the DUT should indicate the reception of an errored packet.
d. In step 6, the DUT should indicate the reception of an errored packet.

10.1.19 Test 3.4.12 – Receive State Diagram - BAD ESD2 State

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send preamble and valid frame data followed by ESD1 and the Ternary symbol for 000 twice.
3. Monitor the DUT to determine if the DUT accepted the packet.
4. Repeat steps 1-3, except instruct the test station to send preamble and valid frame data followed by ESD1, the Ternary symbol for 000, and ESD3 in step 2.
5. Repeat steps 1-3, except instruct the test station to send preamble and valid frame data followed by ESD1, the Ternary symbol for 000, and SSD in step 2.

Observable Results:

a. In step 2, the DUT should indicate the reception of an errored packet.
b. In step 4, the DUT should indicate the reception of an errored packet.
c. In step 5, the DUT should indicate the reception of an errored packet.

10.1.20 Test 3.4.13 – Receive State Diagram - BAD END and RX ERROR states

Procedure:

1. Configure the test stations such that the DUT establishes a valid link while the test station is sending valid Idle.
2. Instruct the test station to send preamble and valid frame data followed by ESD1, ESD2, and ERR_ESD3.
3. Monitor the DUT to determine if the DUT accepted the packet.
4. Repeat steps 1-3,except instruct the test station to send preamble and valid frame data followed by ESD1, ESD2, and all ternary symbols that do not represent ESD3 or ERR_ESD3 in step 2.
5. Repeat steps 1-3, except instruct the test station to send preamble and valid frame data followed by ESD1, ESD2, and SSD in step 2.

Observable Results:

a. In step 2, the DUT should indicate the reception of an errored packet.
b. In step 4, the DUT should indicate the reception of an errored packet.
c. In step 5, the DUT should indicate the reception of an errored packet.

10.2 Appendix 3.B – Test Stations

Purpose: To provide the requirements of the test stations used during 100BASE-T1 PCS testing.

Discussion:

Two test stations will be required to perform all tests that are specified in this document. The 100BASE-T1 Receive Test Station will examine the transmissions from the DUT, and the 100BASE-T1 Transmit Test Station will transmit the necessary test patterns to test the receiver of the DUT. It is possible to combine both stations into one setup.

The 100BASE-T1 Receive Test Station will consist of an oscilloscope and software to capture and decode the transmissions from the DUT. The DUT will connect to the test station through the Line Tap as specified in appendix 3.B. The software will download the capture from the oscilloscope and decode the ternary symbols, using knowledge of the 100BASE-T1 encoding, to create the MII data stream. The test setup is shown in Figure A.1. Note that the MII test station is mandatory for some tests, while it can be replaced with higher layers or a loopback in some tests. Other solutions, such as an FPGA that can capture the ternary symbols, are possible however the test station must not modify or affect the transmissions from the DUT in any manner.

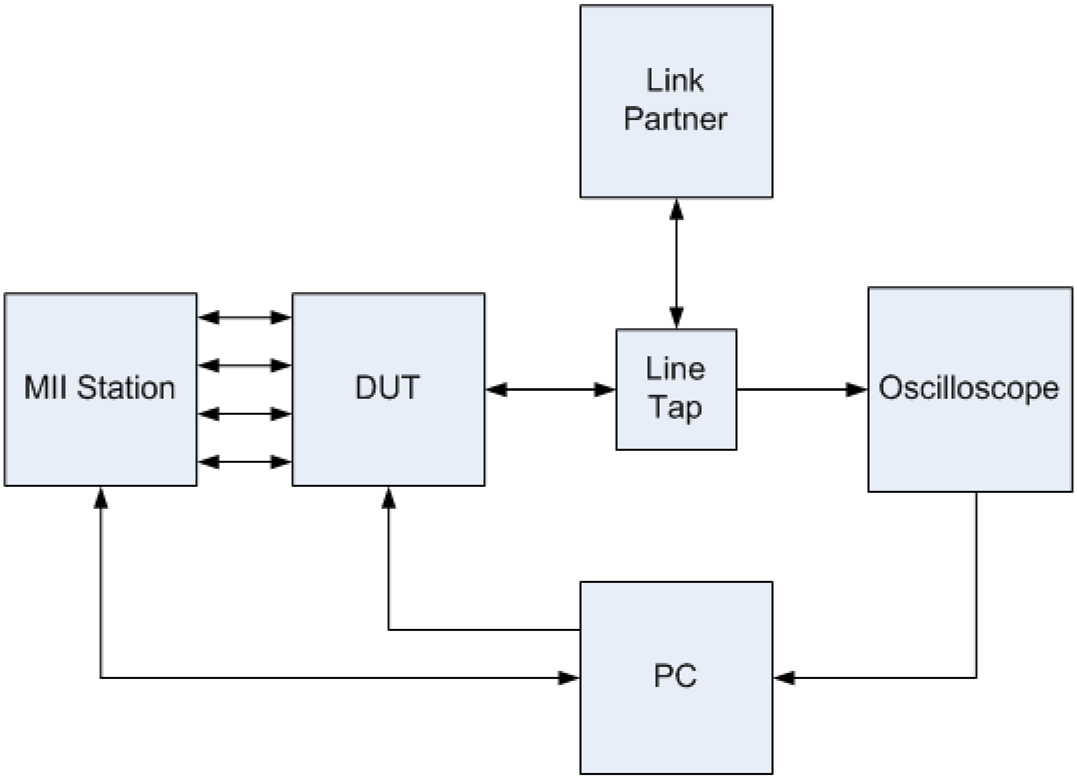

Figure A - 1: 100BASE-T1 Receive Station Setup

The 100BASE-T1 Transmit Test Station will consist of software and hardware that is capable of transmitting arbitrary ternary symbols to the DUT. The ability to send arbitrary sequences, such as invalid transitions of the PCS Transmit State Machine, is essential to fully test the receiver of the DUT.

The test setup is shown in Figure A - 2. Note that the MII test station is mandatory for some tests, while it can be replaced with higher layers or a loopback in some tests.

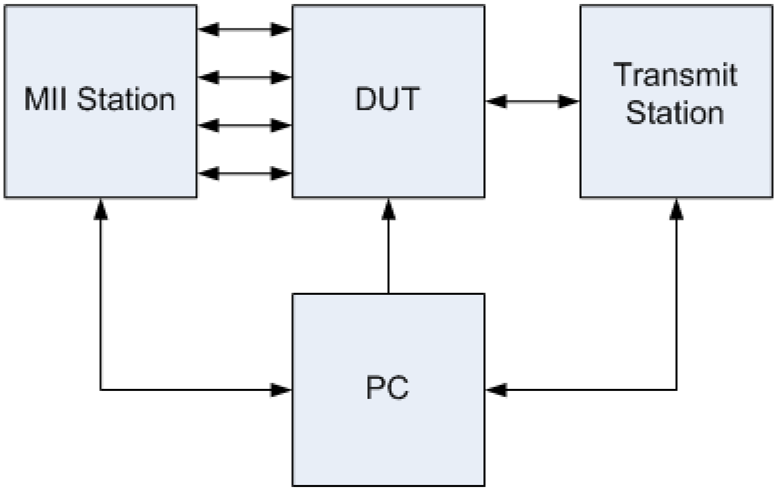

Figure A - 2: 100BASE-T1 Transmit Station Setup

10.3 Appendix 3.C – Line Tap

Purpose: To provide the requirements of a line tap that will be used, in conjunction with an oscilloscope, to capture the transmissions from the DUT.

Discussion: The line tap fixture used for collecting the necessary oscilloscope captures to perform this testing is not specified in this document. Since the line tap is part of the test channel between the Device Under Test (DUT) and Link Partner, the only performance requirement is that the channel (including the line tap) meets the link segment requirement of IEEE Std 802.3bw-2015. However, for information regarding the UNH-IOL's implementation of the line tap used during 100BASE-T1 PCS testing, please contact the UNH-IOL.