SAS Sample Test Suite

Software Version 5.10
Document Version 5.10

November 2012
Teledyne LeCroy Protocol Solutions Group

Trademarks and Servicemarks:
Teledyne LeCroy, Sierra M6-4, Sierra M6-2, STX A6-4, STX M6-1.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Intel and Pentium are registered trademarks of Intel Corporation.

All other trademarks and registered trademarks are property of their respective owners.

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL INFORMATION, EXAMPLES AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE REPRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS ARE FULLY RESPONSIBLE FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN INFORMATION THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT TELEDYNE LECROY FOR A COPY.

Teledyne LeCroy © 2012. All rights reserved.

This document may be printed and reproduced without additional permission, but all copies should contain this copyright notice.

WEEE Program

Teledyne LeCroy
3385 Scott Blvd.
Santa Clara, CA 95054
TEL: 800-909-7112 (USA and Canada)
TEL: 408-6531260
Contents

Introduction .................................................................................................................................. 6

Scope of document.................................................................................................................. 6

Installation, Configuration and Operation ........................................................................... 6

SAS Environment Directory and File .................................................................................. 6

Target Device Configuration Requirements ....................................................................... 7

ATS Selection or Manual Operations ............................................................................... 8

Using ATS to run multiple tests .......................................................................................... 8

Running Individual Test Manually ...................................................................................... 8

Test Script Result Values ....................................................................................................... 9

SAS Test Description ............................................................................................................ 10

Diagram format ..................................................................................................................... 10

Diagram legend .................................................................................................................... 10

Normal Recovery (Test Group A) ...................................................................................... 13

Link Reset (TestA.2 - TestA.3) ............................................................................................. 13

With Identify error recovery (TestA.3) .............................................................................. 13

Error Conditions – Frames Sent to Initiator (Test Group B) ............................................. 14

Errors on XFER_RDY frames (TestB.1 & TestB.2) ............................................................... 14

NAK response from Initiator (TestB.1) .............................................................................. 14

ACK/NAK Timeout detected after sending frame (TestB.2) .................................................. 15

NAK response from Initiator (TestB.3) .............................................................................. 17

ACK/NAK Timeout detected after sending frame (TestB.4) .................................................. 19

Errors on RESPONSE frames (TestB.5 & TestB.6) ........................................................... 21

NAK response from Initiator (TestB.5) .............................................................................. 21

ACK/NAK Timeout detected after sending frame (TestB.6) .................................................. 23

Error Conditions – Frames Received from Initiator (Test Group C) .................................. 25

Errors on COMMAND frames (TestC.1, TestC.2, TestC.3.1-TestC.3.3, TestC.4.1-TestC.1) ................................................................................................................................. 25

CRC or other error resulting in NAK to Initiator (TestC.1) .................................................... 25

WRITE Command Receive DONE(ACK/NAK TIMEOUT) after sending ACK to Initiator (TestC.2)......................................................................................................................... 27

READ Command Receive Error, Insertion & Collision sequenses (TestC.3.1 – TestC.3.3) ................................................................................................................................. 29

New Test for lost DUT ACK for a Read Command (TestC.3.1) (Previously known as TestC.3) ................................................................................................................................. 29

SCSI Command Receive DONE(ACK/NAK TIMEOUT) after sending ACK to Initiator (TestC.4.2) ................................................................................................................................. 31
New Test for lost DUT ACK for a SCSI INQUIRY Command (TestC.4.2) ........................................ 31
Error Recovery on DATA sequence (TestC.5.1-2) .......................................................... 33
CRC or other error resulting in NAK to Initiator (TestC.5.1) ............................................ 33
CRC on data resulting in NAK and abort Response to Initiator (TestC.5.2) ....................... 35
Receive DONE(ACK/NAK TIMEOUT) after sending ACK to the Initiator (TestC.6) ............ 37
Errors on TASK frames (TestC.7 & TestC.8) ................................................................. 39
CRC or other error resulting in NAK to Initiator (TestC.7) ................................................. 39
Receive DONE(ACK/NAK TIMEOUT) after sending ACK to Initiator (TestC.8) .................. 41
SSP_FRAMES - Invalid Additional CDB Length value field (TestC.9.3) ......................... 43
SSP_FRAMES – Missing CDB in SCSI Command (TestC.9.4) .......................................... 43
SAS2r14b, 9.2.5.3 ............................................................................................................... 44
Test Update ....................................................................................................................... 44
Unrecognized or unsupported frames (TestC.10) ............................................................. 46
Errors on Data Offset sequences (TestC.16 - TestC.18) ............................................ 47
SSP_FRAMES – Data Offset Sequence error Missing frame (TestC.16) ......................... 47
SSP_FRAMES – Data Offset Sequence error Out of Order sequence frame (TestC.18) .... 48
Write Data Request with data framing errors (TestC.19) ............................................ 50
SSP_DATA_IU – Write request with Data Timeout response (TestC.19) .......................... 50
Invalid Hashed Address detection (TestC.23.1) .............................................................. 52
SSP ERROR HANDLING – Invalid Hashed Address error detection (TestC.23.1) .......... 52
Command and/or Task Tag Error Detection and Response (TestC.26.1) ....................... 54
SSP ERROR HANDLING – Duplicate Command Tag recovery sequence (TestC.26.1) .... 54
Connection Errors (Test Group D) .................................................................................. 55
Credit Timeout (1 ms) (TestD.1) ........................................................................................ 55
DONE Timeout (TestD.2) ................................................................................................. 57
CLOSE Timeout (TestD.3) ............................................................................................... 59
Receive CREDIT_BLOCKED primitive (TestD.4) .......................................................... 61
Open Timeout (TestD.5) ................................................................................................. 63
Receive BREAK redundant primitive (TestD.7.1 and TestD.7.2) ...................................... 65
SSP ERROR HANDLING – Disabled BREAK_REPLY and Receive BREAK redundant primitive (TestD.7.1) ................................................................. 65
SSP ERROR HANDLING – Enable and verify BREAK_REPLY and Receive BREAK redundant primitive (TestD.7.2) ...................................................... 67
BREAK TIMEOUT after sending BREAK redundant primitive (TestD.8) ......................... 69
Receive BROADCAST(CHANGE) redundant primitive (TestD.9) ................................. 71
BREAK during Write or Read Sequence recovery (TestD.10.2 - TestD.10.3) .................... 72
SSP ERROR HANDLING – Break on Read Data frame sequence (TestD.10.2) .............. 72
SSP ERROR HANDLING – Break on Write Data frame sequence (TestD.10.3) .............. 74
Error Conditions – Connection not Established (Test Group E) ..................................... 76
Receive OPEN_REJECT (Retry class) (TestE.1) .............................................................. 76
Receive OPEN_REJECT (abandon class) (TestE.2) ......................................................... 78
Receive OPEN_REJECT for unsupported protocol type ................................................. 80
Behavior After Receiving a Task Management Request (Test Group F) ....................... 82
ABORT TASK (TestF.1) ................................................................................................. 82
ABORT TASK SET (TestF.2) ......................................................................................... 84
CLEAR ACA (TestF.3) ................................................................................................. 86
CLEAR TASK SET (TestF.4) ......................................................................................... 88
I_T NEXUS RESET (TestF.5) ....................................................................................... 90
LOGICAL UNIT RESET (TestF.6) ................................................................. 92
QUERY TASK (TestF.7) ............................................................................ 94
QUERY TASK SET (TestF.8) ..................................................................... 96
QUERY ASYNCHRONOUS EVENT (TestF.9) ............................................. 98
Task Management Function with Undefined and Reserved command
(TestF.10 & TestF.11) ............................................................................... 100
  SCSI CDB – Undefined Task Management Function (TestF.10) .............. 100
  SCSI CDB – Reserved Task Management Function (TestF.11) .................. 100
TMF or Commands w/Tag conflict during Read or Write command
sequence (TestF.15.2 – TestF.15.5) ........................................................... 102
  SCSI CDB – Identical Write command sequence with TAG Conflict (no retran bit)
  (TestF.15.2) ............................................................................................ 102
  SCSI CDB – Identical Read command sequence with TAG Conflict (no retran
  bit)(TestF.15.3) ........................................................................................ 104
  SCSI CDB – Read and ABORT TASK command sequence with TAG Conflict (TestF.15.4) .... 106
  SCSI CDB – Write and Query TASK command sequence with same TAG Conflict
  (TestF.15.5) ............................................................................................ 108
Introduction

Scope of document

This document describes the SAS Host Interface validation test requirements for error and recovery situations of the Target device (DUT). These tests are an extension of the standard SAS evaluation tests defined by the University of New Hampshire InterOperability Laboratory (UNH IOL) and the Serial Attached SCSI (SAS) Consortium. The details of the UNH IOL tests are not defined in this document. Please refer to the UNH IOL distributed reference documentation for the details of the tests.

Also, this document describes additional behavior expected from the Target device in error and recovery situations where the broader SAS standard permits more than one valid behavior (e.g., abort or retry a NAK’ed frame). The scope for this behavior is restricted to the Application, Transport and Link behavior.

For each of the additional expected behavior, a validating test sequence is defined. These tests will be required in addition to the UNH IOL suite to complete the validation of the Target Devices.

We assume that the reader has some familiarity with the T10 SCSI and SAS standards specification.

Installation, Configuration and Operation

SAS Environment Directory and File

The SAS Test files will reside under the directory “SASTarget”. The directory can be placed in any location on the system. All Scripts and generated Traces files will reside under this directory.

- ATS directory will be used by the Automated Test Suite(ATS) application to run the Trainer test scripts for Target device results.
- ATS/Results directory will contain the ATS Output.txt and HTML Results file, if the operator configures the ATS setting to save those file at this location.
• Projects directory contain the SAS test project file for the Analyzer M6-4 and M6-2. The ATS scripts will determine the Hardware model in use and select the correct project file to use.
• Traces directory will contain the all the Trace (.scs) files generate by the tests.
• Trainer directory contains all the SAS Test Script created for the Trainer Application. An corresponding ATS script will load and start the test using ATS application

Target Device Configuration Requirements

There is a file that requires the operator to modify with a simple text editor (Notepad) before using the Test Scripts. The file “./SASTarget/Trainer/include/DeviceSetting.inc” contains configuration items describing the Testing Environment and the supported function of the Target Device. Each Trainer definition sets the Test Script operations.

• Set Speed = LINK_SPEED_6G (default)
  o LINK_SPEED_1_5G
  o LINK_SPEED_3G
  o LINK_SPEED_6G

• Const SAS_AF_RATE_DEFAULT = 0xA (default)(must match Speed)
  o 0x8 for SAS_AF_RATE_1_5_GBPS
  o 0x9 for SAS_AF_RATE_3_GBPS
  o 0xA for SAS_AF_RATE_6_GBPS

• Const SAS_AF_FRAMING_DEFAULT = 0x0 (default)
  o 0x0 for SAS_AF_512_BYTES
  o 0x1 for SAS_AF_520_BYTES
  o 0x2 for SAS_AF_528_BYTES

• Const SAS_BREAK_REPLY_DEFAULT = 0x01 (default)
  o 0x00 for SAS_ID_BREAK_ONLY (BREAK/BREAK)
  o 0x01 for SAS_ID_BREAK_REPLY (BREAK/BREAK_REPLY)

• Const SAS_INITIATOR_RESPONSE_TMO_DEFAULT = 0x01 (default)
  o 0x00 for INITIATOR_RESPONSE_TMO_DISABLE
  o 0x01 for INITIATOR_RESPONSE_TMO_ENABLE

Optional (Rarely used)
• Set SSC = OFF
• Set SSCType = SSC_DOWN_STREAM
• Set SSCAmplitude = SSC_AMP_1500
ATS Selection or Manual Operations

Using ATS to run multiple tests

To execute the SAS Test Script with the Automated Test Suite, the operator must setup the application to reference the SAS Test environment.
As an example below for ATS, the SAS Test Scripts was located in the system Program Files under the LeCroy SAS Analyzer.

Using the ATS GUI to select the test to run, start the selected test.

Running Individual Test Manually

To run an individual test script, it will require the operator to use the Analyzer and Trainer GUI. Below are the steps needed to run it manually.

1. Start Analyzer and Connect to the Hardware you wish to use. Hardware wiring is Target Port 1 to Target device and Power cable to device as well.
2. Open “Configuration” pull down and select the “Software Setting” and update the “User path:” field with the the location of the SAS Test Scripts and save with OK.
3. If a Project file is open and not the SAS Test version, close the Project window. Open a Project file (Pull down “File” select “OPEN” or hit the OPEN button) located in the Project directory. Select the project matching the Hardware connected.
4. Start the Trainer GUI (Pull down “View” select “Show Trainer File” or hit the Trainer button) and configure the port setting by select the Pull down “Setup” and selecting “Port Configuration”. Change port Assignment to AT, -, -, - and save with OK.

5. Open a Trainer script (Pull down “file” select “Open” or Open Button) and select test script from directory Trainer/TestScript.

6. To start the test, initiate the analyzer trace capture (Start Analyzer button (Red Dot button)) then start Generator (Pull down “Generate” select “Start Generate” or hit the Start Generate button). The test is started.

7. View the Trainer Port “Show Port Status” screen (Hit the lower right Green/Red LED looking to open window) to view the capture and trainer status value. Trainer column will display the test results. Stop Analyzer (Red Square button) to view captured trace.

**Test Script Result Values**

The test scripts will return a status value of the following:

- **0. NO RESULT**: System error condition discovered with selected Test.
- **1. PASSED**: Test completed with expected results.
- **2. FAILED**: Test completed but with unexpected or invalid SAS events.
- **3. SETUP ERROR**: Test was unable to establish DUT state for proper test condition.

Failed or Setup can be investigated by viewing the captured trace file. Each of the failed tests can be re-tested to see if an internal condition in the DUT could have cleared itself.
SAS Test Description

Diagram format

Each important behavior will be documented by a ladder diagram as well as text. Most of these behaviors pertain to handling errors or exceptional conditions within the context of executing a specific command. A sample diagram is shown below.

It displays three levels of the testing platform. First, the transaction between the over the physical interface between the initiator/Testing Station and the Device Under Test (DUT). Next the Target Primitive Logic, which describes the action taken by the logic developed for primitive dword level action. And finally, the notification event and action that is taken by the Target Protocol following the SAS protocol requirements.

Diagram legend

- **Primitive**
  
  The diagrams are used for visualization of the major functional components, control and data flow of the SSP link. The left column of the diagram pertains to the Initiator functionality. Target functionality is split between center and right columns. Center column reflects blocks and functionality implemented in SSP link logic for Primitives. Right column reflects functionality implemented in Target device SAS protocol logic.

- **SSP Link**
  
  Graphical shapes, used for different blocks and functions, are shown to the left. Arrows are used to show direction of the control and data flow and events generated by SSP HW.
Shape crossed with two lines shows SSP primitive, command or response frame which needs to be deleted to recreate exception case shown in the diagram.

Values of the task management and response frame fields pertaining to the behavior outlined in the diagram are shown adjacent to the shape representing respective frame.

Various behaviors of the SSP command execution and task management allowable within multiple connections or only single connection are not shown by separate diagrams but delineated by the text notes on the diagram.
Figure 1 - Example of the flow chart

- **Initiator/Test Station**
  - Read Command
    - SSP Link
      - ACK Primitive
        - NAK Primitive
          - ACK/NACK Primitive
            - ACK Primitive
              - DONE(ACK/NACK Timeout)
  - ACK/NACK Primitive

- **Target Primitive Logic**
  - SSP Link
    - Response Frame
      - Sense Key=ABORT COMMAND
        - ASC/Q = NAK RECEIVED
          - Retransmit Bit = 1
          - 1 ms Timeout Event
          - Response frame is transmitted in a new connection

- **Target Protocol Logic**
  - SAS Protocol
    - Data Frame
      - Response Frame

Response frame could be transmitted in the same or new connection.
Normal Recovery (Test Group A)

Link Reset (TestA.2 - TestA.3)

With Hard Reset (TestA.2)

Link Reset with Hard Reset results in a full reset of the Logical Unit. This shall be handled by firmware as if a Logical Unit Reset Task Management request has been received.

Test Procedure:

1. The Testing Station is instructed to transmit COMINIT to start a phy Reset sequence with the DUT.
2. The Testing Station is instructed to complete a phy Reset sequence with the DUT.
3. Once the phy reset sequence between the DUT and the Testing Station are complete, the Testing Station is instructed to transmit 6 consecutive HARD_RESET primitives. This must occur before the Identify sequence is complete.

Observable Results: Verify that the DUT transmits COMINIT upon receiving the HARD_RESET primitives from the Testing Station.

With Identify error recovery (TestA.3)

Link Reset with the Test Station issuing no, and bad CRC Identify. The DUT must reset the link in 1 ms to initiate another connection attempt.

Test Procedure:

1. The Testing Station is instructed to transmit COMINIT to start a phy Reset sequence with the DUT.
2. The Testing Station is instructed to complete a phy Reset sequence with the DUT.
3. Once the phy reset sequence between the DUT and the Testing Station are complete, the Testing Station is instructed to test the Identify sequence.
   a) No Identify Frame sent
   b) An Identify Frame with bad CRC before DUT sends Identify
   c) An Identify Frame with bad CRC after DUT sends Identify

Observable Results: Verify that the DUT transmits COMINIT upon 1 ms after sending Identify.

4. Once the phy reset sequence between the DUT and the Testing Station are complete, the Testing Station is instructed to initiate 4 identify frames colliding in time frame with the DUT Identify frame

Observable Results: Verify that the DUT maintain link connection.
Error Conditions – Frames Sent to Initiator (Test Group B)

Errors on XFER_RDY frames (TestB.1 & TestB.2)

NAK response from Initiator (TestB.1)

Target device shall send a Response frame with CHECK CONDITION status, sense key = ABORT COMMAND, and additional sense code = NAK RECEIVED. This Response frame may be sent in the same or in a new connection.

Test Procedure:

1. Power on the DUT.
2. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
3. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI MODE SENSE command for the Disconnect-Reconnect Mode Page. Close the connection.
4. Allow the DUT to open an SSP connection to the Testing Station and transmit a DATA frame with the Mode Page Block Descriptor and a SCSI response frame to the received MODE SENSE command.
5. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE_BUFFER command for 512 bytes. Close the connection.
6. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY or SCSI response frame.
7. The Testing Station is instructed to send a NAK response to the XFER_RDY.

Observable Results: Verify that the DUT transmits a SCSI response Abort with sense key of ABORTED COMMAND and sense code of NAK RECEIVED.
ACK/NAK Timeout detected after sending frame (TestB.2)

Target device shall send a Response frame with Check Condition status, sense key = Aborted Command, and additional sense code = ACK/NAK TIMEOUT. This Response frame shall be sent in a new connection.

Test Procedure:

1. Power on the DUT.
2. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
3. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI MODE SENSE command for the Disconnect-Reconnect Mode Page. Close the connection.
4. Allow the DUT to open an SSP connection to the Testing Station and transmit DATA frame with the Mode Page Block Descriptor and a SCSI response frame to the received MODE SENSE command.

5. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 2048 bytes. Close the connection.

6. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY or SCSI response frame.

7. The Testing Station is instructed to not send ACK/NAK response to the XFER_RDY.

8. Wait for DUT to send DONE (ACK/NAK TIMOUT) after 1 ms. Complete close connection.

9. Allow the DUT to OPEN connect to the Test Station.

Observable Results: Verify that the DUT sends a SCSI Response Abort with sense key of ABORTED COMMAND and sense code of ACK/NAK TIMEOUT.

Figure 3 - ACK/NAK Timeout detected after sending frame
Errors on DATA frames (TestB.3 & TestB.4)

**NAK response from Initiator (TestB.3)**

Target device shall send a Response frame with Check Condition status, sense key = Aborted Command, and additional sense code = NAK RECEIVED. This Response frame may be sent in the same or in a new connection.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI READ(10) command for 1 frame. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit DATA frames. The Testing Station is instructed to transmit a NAK to the DATA frame.

Observable Results: Verify that the DUT transmits a SCSI response Abort with sense key of ABORTED COMMAND and sense code of NAK RECEIVED.
Figure 4 - NAK response from Initiator
**ACK/NAK Timeout detected after sending frame (TestB.4)**

Target device shall send a Response frame with Check Condition status, sense key = Aborted Command, and additional sense code = ACK/NAK TIMEOUT. This Response frame shall be sent in a new connection.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI READ _BUFFER command for N bytes (specify). Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit DATA frame. The Testing Station is instructed not to send ACK/NAK to the DUT DATA frame.
4. Wait for the DUT to timeout (1 ms) and send a DONE (ACK/NAK TIMEOUT).

Observable Results: Verify that the DUT transmit a SCSI response Abort with sense key of ABORTED COMMAND and sense code of ACK/NAK TIMEOUT.
Figure 5 - NAK Timeout detected after sending frame
Errors on RESPONSE frames (TestB.5 & TestB.6)

NAK response from Initiator (TestB.5)

Target device shall resend the frame one time with the Retransmit bit = 1 and all other fields the same as in the first transmission of the frame. (I.e., can’t disable retries for a response frame). The retransmitted Response frame may be sent in the same or in a new connection.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 1024 bytes. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY.
4. The Testing Station is instructed to open a connection to the DUT and transmit 1024 byte data frame. Close the connection.
5. Allow the DUT to open an SSP connection and transmit a SCSI RESPONSE frame to the received command and data. The Testing Station is instructed to transmit NAK to the RESPONSE frame. Second part of test will instruct Testing Station to send no ACK response and wait for DONE ACK/NAK TIMEOUT

Observable Results: Verify that the DUT transmits the Response frame again. It should be on a new connection request but not required.
Figure 6 – NAK and ACK/NAK Timeout response from Initiator
**ACK/NAK Timeout detected after sending frame (TestB.6)**

Target device shall resend the frame once in a new connection with the Retransmit bit = 1 and all other fields the same as in the first transmission of the frame. (i.e., can’t disable retries for a response frame). This Response frame shall be sent in a new connection.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.
2. The Testing Station is instructed to open an SSP connection to the Testing Station.
3. Wait for the DUT to transmit OPEN_ACCEPT and RRDY. The Testing Station is instructed to transmit a SCSI INQUIRY command to the DUT then close the connection.
4. Wait for the DUT to open an SSP Connection to the Testing Station. Allow the DUT to transmit a SCSI Response. The Testing Station is instructed to transmit a NAK.
5. Wait for the DUT to Timeout 1 ms and send a DONE (ACK/NAK TIMEOUT).

Observable Results: Verify that the DUT retransmit Response Frame with Retransmit bit = 1.
Figure 7 - ACK/NAK Timeout detected after sending frame
Error Conditions – Frames Received from Initiator (Test Group C)

Errors on COMMAND frames (TestC.1, TestC.2, TestC.3.1-TestC.3.3, TestC.4.1-TestC.1)

**CRC or other error resulting in NAK to Initiator (TestC.1)**

Target device shall send NAK to the Initiator. The frame is not recognized as a Command Frame. The frame is discarded. If the Initiator resends the COMMAND frame, the Target device will see that as a new command.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.
2. The Testing Station is instructed to open a SSP connection to the DUT and transmit a SCSI_INQUIRY command with an invalid CRC.
3. Wait for the DUT to transmit NAK (CRC ERROR).
4. The Testing Station is instructed to open a SSP connection to the DUT and transmit a SCSI_INQUIRY command with a valid CRC.

Observable Results: Verify that the DUT transmit ACK.
Figure 8 - or other error resulting in NAK to Initiator
WRITE Command Receive DONE(ACK/NAK TIMEOUT) after sending ACK to Initiator (TestC.2)

The Target device shall continue executing the command unless it is aborted by Initiator. If the Initiator does not recognize a XFER_RDY, Data, or Response frame for this I_T_L_Q nexus within the timeout period, then the Initiator should issue a QUERY TASK and may issue an ABORT TASK. (Abort behavior is described elsewhere in this document)

Write sequence (see Figure 9 below for details)
Test Procedure:

1. Power on the DUT.
2. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
3. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI MODE SENSE command for the Disconnect-Reconnect Mode Page. Close the connection
4. Allow the DUT to open an SSP connection to the Testing Station and transmit DATA frame with the Mode Page Block Descriptor and a SCSI response frame to the received MODE SENSE command.
5. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for a single frame write. Wait for the DUT to transmit an ACK. Ignore the ACK and wait 1 ms after sending the WRITE command and transmit a DONE (ACK/NAK TIMEOUT). Close the connection.
6. The Testing Station is instructed to initiate an open SSP connection to the DUT and transmit a TASK MANAGEMENT Frame type QUERY TASK. And, wait for DUT to respond with an ACK.
7. Allow the DUT to transmit an XFER_RDY frame. Which the Testing Station will respond with and ACK.
8. The Testing Station is instructed to send the single Write DATA frame. And, wait for the DUT to respond with an ACK.
9. Wait for the DUT to transmit the following frame, the Read DATA frame, a Response frame for the QUERY TASK and the Read Command Response frame. The order of incoming frame should be in this order. The Testing Station will transmit an ACK response for each frame.

Observable Results: Verify that the DUT completes the final 3 frames.
Figure 9 - Write Receive DONE(ACK/NAK TMO) after sending ACK
READ Command Receive Error, Insertion & Collision sequences (TestC.3.1 – TestC.3.3)

New Test for lost DUT ACK for a Read Command (TestC.3.1) (Previously known as TestC.3)

Read Sequence (see Figure 10 below for details)

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.

2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI READ_BUFFER command for a single DATA frame read.

3. Wait for the DUT to transmit an ACK the Read command and ignore the response. The Testing Station will wait 1 ms after transmitting the command frame. Then, the Testing Station is instructed to send a DONE (ACK/NAK TIMEOUT) and close the connection.

4. The Testing Station is instructed to initiate an open SSP connection to the DUT and transmit a TASK MANAGEMENT Frame type QUERY TASK. And, wait for DUT to respond with an ACK.

5. Wait for the DUT to transmit the following frames, a Response frame for the QUERY TASK and the Read Command Response frame. The order of incoming frame should be in this order. The Testing Station will transmit an ACK response for each frame.

Observable Results: Verify that the DUT completes the final 2 frames.
Figure 10 - Read Receive DONE(ACK/NAK TIMEOUT) after sending ACK
SCSI Command Receive DONE(ACK/NAK TIMEOUT) after sending ACK to Initiator (Test TestC.4.2)

New Test for lost DUT ACK for a SCSI INQUIRY Command (TestC.4.2)

Command Sequence (see Figure 11 below for details)
Test Procedure:
1. Power on the DUT.
2. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
3. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI INQUIRY. Wait for the DUT to response with an ACK and ignore it.
4. The Testing Station will wait 1 ms after sending the Command frame before transmitting a DONE (ACK/NAK TIMEOUT) and close connection.
5. The Testing Station is instructed to open an SSP connection to the DUT and transmit a Task Management frame type QUERY TASK. And, wait for the DUT to respond with an ACK.
6. Wait for the DUT to transmit the following 2 frames, a Command Response DATA frame and the INQUIRY response frame. The order of incoming frame should be in this order. The Testing Station will transmit an ACK response for each frame.

Observable Results: Verify that the DUT completes the final QUERY TASK Response frame.
Figure 11 - INQUIRY Command Receive DONE(ACK/NAK TIMEOUT) after sending ACK
Error Recovery on DATA sequence (TestC.5.1-2)

CRC or other error resulting in NAK to Initiator (TestC.5.1)

The Target device shall send NAK to the Initiator. The Initiator shall send Abort Task to the Target device.

Test Procedure:
1. Power on the DUT.
2. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
3. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI MODE SENSE command for the Disconnect-Reconnect Mode Page. Close the connection.
4. Allow the DUT to open an SSP connection to the Testing Station and transmit DATA frame with the Mode Page Block Descriptor and a SCSI response frame to the received MODE SENSE command.
5. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE_BUFFER command for a number of bytes to generate only a single frame. Close the connection.
6. Wait for the DUT to send ACK to WRITE command
7. Wait for the DUT to send a XFER_RDY frame to Testing Station. And, respond to the RRDY with an ACK.
8. The Testing Station is instructed to send the Data Frame with an invalid CRC.
9. Wait for the DUT to respond with a NAK response.
10. The Testing Station is then instructed to send a Task Management function ABORT TASK. The DUT should send an ACK response to the Command.

Observable Results: Verify that the DUT transmit the ABORT TASK Response frame.
Figure 12 - CRC or other error resulting in NAK to Initiator
**CRC on data resulting in NAK and abort Response to Initiator (TestC.5.2)**

The Target device shall send NAK to the Initiator. The Target device shall send a response with Check Condition, Abort Task and reason NAK received to the initiator.

Test Procedure:

1. Power on the DUT.

2. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.

3. The Testing Station is instructed to send a READ10 command for multiple data frame request length.

4. Allow the DUT to initiate a connection which the Test Station will accept. And, the Test Station will make several RRDY available to the DUT to send data frames.

5. The Testing Station will send an ACK, NAK (CRC ERROR), ACK and NAK (CRC ERROR) responses for the first 4 DATA frame it receives.

**Observable Results:** Verify that the DUT transmit the CHECK CONDITION, ABORT TASK with ASC/Q of NAK RECEIVED Response frame.

6. The Testing Station is instructed to send a READ10 command for large multiple data frame request length.

7. Allow the DUT to initiate a connection which the Test Station will accept. And, the Test Station will make several RRDY available to the DUT to send data frames.

8. Allow the DUT to initiate a connection which the Test Station will accept. And, the Test Station will make several RRDY available to the DUT to send data frames.

9. The Testing Station will send an ACK, NAK (CRC ERROR), ACK responses for the first 2 DATA frames it receives. It will continue with ACK & RRDY until completed.

**Observable Results:** Verify that the DUT transmit the CHECK CONDITION, ABORT TASK with ASC/Q of NAK RECEIVED Response frame.
Figure 13 – Delayed NAK CRC on data resulting in abort Response to Initiator
Receive DONE(ACK/NAK TIMEOUT) after sending ACK to the Initiator (TestC.6)

Because retries are to be disabled, the Initiator cannot resend the Data frame. The Target should send a Response frame terminating the command with CHECK CONDITION status with ASC/ASCQ = 4B03h, ACK/NAK Timeout. This Response frame shall be sent in a new connection.

Test Procedure:

1. Power on the DUT.
2. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
3. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 1024 bytes. Close the connection.
4. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY.
5. The Testing Station is instructed to open a connection to the DUT and transmit the 1024 byte data frame.
6. Wait for DUT to transmit ACK to DATA Frame. The Testing Station is instructed to ignore the ACK and wait 1 ms. The Test Station is instructed to transmit a DONE (ACK/NAK TIMEOUT) to the DUT.

Observable Results: Verify that the DUT transmit a SCSI response is either 1) STATUS is GOOD or 2) CHECK_CONDITION ACS/ACSQ (4B03h) and code to ACK/NAK TIMEOUT.
Figure 14 - Receive DONE(ACK/NAK TIMEOUT) after sending ACK
Errors on TASK frames (TestC.7 & TestC.8)

CRC or other error resulting in NAK to Initiator (TestC.7)

The Target shall send NAK to the Initiator. The frame is not recognized as a Task Frame and is discarded.

The Initiator should resend the Task frame with the Retransmit bit set to one. The Target will see this as a new request.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.

2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a TASK MANAGEMENT command with an Invalid CRC.

3. Wait for the DUT to transmit NAK then close the connection. The Test Station is instructed to connect to the DUT and transmit a valid TASK MANAGEMENT command with the RETRANSMIT bit enabled.

4. Wait for the DUT to transmit an ACK

Observable Results: Verify that the DUT transmit with a TASK MANAGEMENT Response frame to the command with RESPONSE_DATA and TMF_COMPLETE.
Figure 15 - CRC or other error resulting in NAK to Initiator
Receive DONE(ACK/NAK TIMEOUT) after sending ACK to Initiator (TestC.8)

The Target device shall continue executing the task.

The Initiator may resend the Task frame with the Retransmit bit set to one.

Test Procedure:
1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a valid TASK MANAGEMENT command.
3. Wait for the DUT to transmit ACK. The Test Station is instructed to close connect with a DONE(ACK/NAK TIMEOUT)
4. The Testing Station will connect with the DUT and transmit the TASK MANAGEMENT command with the RETRANSMIT bit enabled.
5. Wait for the DUT to transmit an ACK

Observable Results: Verify that the DUT transmit with a TASK MANAGEMENT Response frame to the command.
Figure 16 - Receive DONE(ACK/NAK TIMEOUT) after sending ACK
Invalid frames for SSP_FRAMES (TestC.9.1 – TestC.9.8)

An invalid frame is one that has a valid SAS frame type but which has an error in length, content, or occurrence.

Examples include:
- A frame shorter than the minimum defined length for that frame type (Test 9.1.17)
- A Command or Task frame that has the Target Port Transfer Tag set to a value other than FFFFh (Test 9.1.18)
- A Response frame delivered to a Target (Test 9.1.19)

When a Target receives an invalid frame, it shall reply with a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID FRAME.

**SSP_FRAMES - Invalid Additional CDB Length value field (TestC.9.3)**

Test Procedure:
1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a normal SCSI INQUIRY command. And wait for DUT to respond with good status.
3. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI INQUIRY command with an invalid setting of 3Fh for the Additional CDB Length field. Close the connection.
4. Allow the DUT to open an SSP connection and transmit a SCSI response frame to the received command and data.

Observable Results: Verify if the DUT sets the DATAPRES field to RESPONSE_DATA and the RESPONSE CODE field to INVALID FRAME.

**SSP_FRAMES – Missing CDB in SCSI Command (TestC.9.4)**

Test Procedure:
1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a normal SCSI INQUIRY command. And wait for DUT to respond with good status.
3. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI command with a missing CDB portion of the frame. Close the connection.
4. Allow the DUT to open an SSP connection and transmit a SCSI response frame to the received command and data.

Observable Results: Verify if the DUT sets the DATAPRES field to RESPONSE_DATA and the RESPONSE_CODE field to INVALID FRAME.

SSP_FRAMES – Variable Length Command missing additional Data (TestC.9.5)

*SAS2r14b, 9.2.5.3*

*Test Update*
Test 9.1.17x

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a normal SCSI INQUIRY command. And wait for DUT to respond with good status.
3. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI command with a variable format CDB and the additional data portion of the frame missing. Close the connection
4. Allow the DUT to open an SSP connection and transmit a SCSI response frame to the received command and data.

Observable Results: Verify if the DUT sets the DATAPRES field to RESPONSE_DATA and the RESPONSE_CODE field to INVALID FRAME.
Invalid Frame

ACK Primitive

SSP Link

ACK Primitive

Response Frame

Status = CHECK_CONDITION
Response Code = INVALID_FRAME

SAS Protocol
**Unrecognized or unsupported frames (TestC.10)**

An unrecognized frame is one with an unrecognized frame type.

An unsupported frame (for a target) is a XFER_RDY and Response frame. A target-only device is not required to distinguish between these two situations.

An unrecognized or unsupported frame shall be discarded by the Target.

**Test Procedure:**

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.

2. The Testing Station is instructed to open an SSP connection to the DUT and transmit an unsolicited SCSI Response Frame XFER_RDY to DUT.

**Observable Results:** Verify that the DUT send an ACK the incoming frame and discards the frame with no additional action.

3. The Testing Station is instructed to open an SSP connection to the DUT and transmit an unsolicited SCSI RESPONSE Framew/Status = OK to DUT.

**Observable Results:** Verify that the DUT send an ACK the incoming frame and discards the frame with no additional action.

---

**Figure 17 - Unrecognized or unsupported frames**

```
+----------------+                +----------------+                +----------------+
|  Initiator/Test Station |      | Target Primitive Logic |      | Target Protocol Logic |
|-----------------|                |-----------------|                |-----------------|
| Invalid Frame |                |        SSP Link       |                |       SAS Protocol        |
|                 |                |        ACK Primitive   |                | Frame is ignored and discarded |
+----------------+                +----------------+                +----------------+
```
Errors on Data Offset sequences (TestC.16 - TestC.18)

The data offset sequence can possible be altered through the network of expanders. Data sequencing for the disk target device does not support any order sequencing problems. The Write request must be aborted.

**SSP_FRAMES – Data Offset Sequence error Missing frame (TestC.16)**

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 2048 bytes. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY.
4. The Testing Station is instructed to open a connection to the DUT and transmit the 2048 bytes of data. Send 3 DATA frames of 512 bytes each with Offset 0000, 0200 and 0600.

Observable Results: Verify that the DUT responds to the write data sequence with a Response of CHECK CONDITION and ASC/Q = 0x4B05 (Data Offset Error)

**SSP_FRAMES – Data Offset Sequence error Duplicate frame (TestC.17)**

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 2048 bytes. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY.
4. The Testing Station is instructed to open a connection to the DUT and transmit the 2048 bytes of data.
5. Send 5 DATA frames of 512 bytes each with the following. Repeat step 2 – 5 for each sequence.
   a. Offset 0000, 0000, 0200, 0400 and 0600.
b. Offset 0000, 0200, 0200, 0400 and 0600.
c. Offset 0000, 0200, 0400, 0400 and 0600.
d. Offset 0000, 0200, 0400, 0600 and 0600.(optional)

Observable Results: Verify that the DUT responds to the Write data sequence with a Response of CHECK CONDITION and ASC/Q = 0x4B05 (Data Offset Error)

**SSP_FRAMES – Data Offset Sequence error Out of Order sequence frame (TestC.18)**

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.

2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 2048 bytes. Close the connection

3. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY.

4. The Testing Station is instructed to open a connection to the DUT and transmit the 2048 bytes of data. Send 5 DATA frames of 512 bytes each with Offset 0000, 0600, 0200, 0400 and 0600.

Observable Results: Verify that the DUT responds to the write data sequence with a Response of CHECK CONDITION and ASC/Q = 0x4B05 (Data Offset Error)
Figure 18 - Data Offset sequence error (Missing, Duplicate, Out of Order)
Write Data Request with data framing errors (TestC.19)

SSP_DATA_IU – Write request with Data Timeout response (TestC.19)

An initiator that request a Write but does not send any data will require the DUT to initiate an Abort to the command.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 2048 bytes. Close the connection
3. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY.
4. The Testing Station is instructed to wait for the DUT to timeout and send response.

Observable Results: Verify that the DUT responds to the write data sequence with a CHECK CONDITION and ASC/Q = 0x4B06 (Initiator Response Timeout)
Figure 19 - Write Request w/no data sent Timeout sequence
Invalid Hashed Address detection (TestC.23.1)

**SSP ERROR HANDLING – Invalid Hashed Address error detection (TestC.23.1)**

An invalid Source or Destination hashed address error condition could be detected. SAS version 2.0 specifies this as a “may check” optional feature. With expanders in larger and larger SAS networks, these fields will become more relative as the possibility of frame mis-routing increases.

Test Procedure:
1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 512 bytes (1 frame). Then, close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY.
4. The Test Station is instructed to open an SSP connection to the DUT and transmit a DATA frame with an invalid Source Hashed Address. Then, close the connection.

Observable Results: Verify that the DUT discards the frame. The DUT may wait for the correct data and possibly timeout and ABORT the connection. OR, may send a vendor-specific confirmation to the SCSI application layer to cause the command using that initiator port transfer tag to be aborted.

6. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
7. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 512 bytes (1 frame). Then, close the connection.
8. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY.
9. The Test Station is instructed to open an SSP connection to the DUT and transmit a DATA frame with an invalid Destination Hashed Address. Then, close the connection.

Observable Results: Verify that the DUT discards the frame. The DUT may wait for the correct data and possibly timeout and ABORT the connection. OR, may send a vendor-specific confirmation to the SCSI application layer to cause the command using that initiator port transfer tag to be aborted.
Figure 20 - Invalid Hashed Address error detection

Status: CHECK CONDITION
ASC = 0xB (Aborted Command)
ASC/Q = 0x4B06 (Initiator Response Timeout)  OR
Status: CHECK CONDITION
ASC = 0xB (Aborted Command)
ASC/Q=Vendor Specific (Immediate response to detection)
Command and/or Task Tag Error Detection and Response (TestC.26.1)
SAS target ports shall.

SSP ERROR HANDLING – Duplicate Command Tag recovery sequence (TestC.26.1)

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. Test Station is instructed to initiate a WRITE10 command to DUT and wait for DUT XFER_RDY response.
3. The Test Station is then instructed to send 2 additional WRITE10 Commands using the same TAGs not of the first write command.
4. And, the Test Station is then instructed to send the DATA frame from the first write request.

Observable Results: Verify that the DUT responds the proper RESPONSE frame CHECK CONDITION with Response Code set to OVERLAPPING INITIATOR PORT TRANSFER ATTEMPTED.
Connection Errors (Test Group D)

Credit Timeout (1 ms) (TestD.1)

If the Target device is connected, has a transmit credit balance of zero, has one or more frame to send, and has not received an RRDY credit for 1 ms, then the Target device shall send a DONE (CREDIT TIMEOUT) primitive. The Initiator should send DONE/CLOSE and the Target device should send CLOSE to complete closing the connection.

Test Procedure:
1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT.
3. Wait for the DUT to transmit OPEN_ACCEPT and RRDY. The Testing Station is instructed to transmit a SCSI INQUIRY command to the DUT then close the connection.
4. Wait for the DUT to open an SSP connection to the Testing Station. The Testing Station is instructed to not transmit RRDY to grant credit for the DUT to transmit a SCSI Response to the Testing Station.

Observable Results: Verify that the DUT transmitted DONE (CREDIT TIMEOUT) within 1 ms of opening the SSP connection.

Test Update

Test Procedure:
1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI READ_BUFFER command for 2 512 byte frames. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit the first DATA frame with a RRDY from the Testing Station. The Testing Station is instructed to transmit an ACK for the DATA frame but not transmit a RRDY for the next DATA frame.

Observable Results: Verify that the DUT timeout (1 ms) for the RRDY. The DUT will initiate a DONE (CREDIT TIMEOUT) response and CLOSE.
Figure 21 - Credit Timeout

Target has Data Frame to transmit but no RRDY credits left

1 ms Timeout Notification

1 ms Timer

DONE (CREDIT TIMEOUT)

CLOSE

Initiator/Test Station

Target Primitive Logic

Target Protocol Logic

RRDY

ACK

CLOSE

SAS Protocol

SAS Protocol

Data Frame
DONE Timeout (TestD.2)

If, after sending DONE or DONE (CREDIT TIMEOUT), the target does not receive DONE, or a frame, within 1 msec, then a DONE timeout condition exists. The Target device shall transmit a BREAK primitive sequence.

Test Procedure:
1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT.
3. Wait for the DUT to transmit OPEN_ACCEPT and RRDY. The Testing Station is instructed to transmit a READ BUFFER of 1024 byte to the DUT then close the connection.
4. Wait for the DUT to open an SSP connection to the Testing Station. The Testing Station is instructed to one transmit RRDY to grant credit for the DUT to transmit a SCSI DATA frame to the Testing Station. Test Station will ACK but not send a DONE in response to DUT DONE.
5. Wait 1 ms for the DUT to send the BREAK (6 primitives) and instruct the Testing Station to reply with BREAK.

Observable Results: Verify that the DUT transmits a BREAK primitive sequence after a 1 ms timeout.
Figure 22 - DONE Timeout
CLOSE Timeout (TestD.3)

If the Target device is the first in a connection to send CLOSE, then it shall wait for 1 ms to receive CLOSE in response. If it does not receive CLOSE within 1 ms, the Target device shall transmit a BREAK primitive sequence.

Test Procedure:
1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT.
3. Wait for the DUT to transmit OPEN_ACCEPT and RRDY. The Testing Station is instructed to transmit a READ BUFFER of 1024 bytes to the DUT then close the connection.
4. Wait for the DUT to open an SSP connection to the Testing Station. The Testing Station is instructed to one transmit RRDY to grant credit for the DUT to transmit a SCSI DATA frame to the Testing Station. Test Station will ACK send a DONE in response to DUT DONE. But, it will not send a CLOSE to the DUT CLOSE
5. Wait 1 ms for the DUT to send the BREAK (6 primitives) and instruct the Testing Station to reply with BREAK.

Observable Results: Verify that the DUT transmits a BREAK primitive sequence after a 1 ms timeout.
Figure 23 - CLOSE Timeout
Receive CREDIT_BLOCKED primitive (TestD.4)

The Target device shall send any outstanding frames for which it has credits, then send DONE. After the connection is closed, if the Target device has more frames to send, it shall open a new connection and attempt to send them.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.

2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI READ_BUFFER command for 2 512 byte frames. Close the connection

3. Allow the DUT to open an SSP connection to the Testing Station and transmit the first DATA frame with a RRDY from the Testing Station.

4. The Testing Station is instructed to transmit an ACK for the DATA frame. Then the Testing Station will transmit a CREDIT_BLOCK primitive for this connection.

Observable Results: Verify that the DUT does not send the next DATA frame. And, the DUT sends a DONE before closing the connection. The DUT should then transmit an OPEN to establish a new connection for the remaining DATA frame.
Figure 24 - Receive CREDIT_BLOCKED primitive
Open Timeout (TestD.5)

When a target attempts to open a connection to an initiator in order to send a frame, the initiator is required to respond with OPEN_ACCEPT or some form of OPEN_REJECT within 1 ms. If no such response is received, the target recognizes an Open timeout condition. The target shall check the I_T Nexus Loss timer, and start it if it is not already started. If the timer has already been started, it shall continue to run. See the next Connection Errors section for the response to having the timer expire. The target shall retry the connection request until it either receives OPEN_ACCEPT or the I_T Nexus Loss timer expires.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT.
3. Wait for the DUT to transmit OPEN_ACCEPT and RRDY. The Testing Station is instructed to transmit a SCSI INQUIRY command to the DUT then close the connection.
4. Wait for the DUT to open an SSP connection to the Testing Station. The Testing Station is instructed not to respond the OPEN or any subsequent OPEN request.

Observable Results: Verify that the DUT re-transmits the OPEN request until I_T Nexus Loss timeout.
Figure 25 - Open Timeout
**Receive BREAK redundant primitive (TestD.7.1 and TestD.7.2)**

When the drive receives a BREAK redundant primitive, it immediately closes the connection, if one is open, on the receiving port.

The drive responds to the BREAK with a BREAK_REPLY redundant primitive if the BREAK_REPLY method of response is enabled. If the BREAK_REPLY method of response is not enabled, the drive responds to BREAK with a BREAK redundant primitive.

**SSP ERROR HANDLING – Disabled BREAK_REPLY and Receive BREAK redundant primitive (TestD.7.1)**

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.
2. The Testing Station is instructed to transmit an Open Address frame to the DUT, followed by BREAK primitives (6 times).

Observable Results: Verify that the DUT transmits BREAK primitive (6 times) response sequence within the 1 ms break timeout period.

3. The Testing Station is instructed to transmit an Open Address frame to the DUT. Test station will wait for DUT to accept Open request.
4. Wait for DUT to transmit BREAK primitive (6 times) due to connection idle timeout 1 ms.
5. The Test Station will transmit BREAK_REPLY primitive (6 times) as an invalid response.

Observable Results: Verify that the DUT does nothing and wait for 1 ms Break response timeout.

6. The Testing Station is instructed to transmit an Open Address frame to the DUT. Test station will wait for DUT to accept Open request.
7. Wait for DUT to transmit BREAK primitive (6 times) due to connection idle timeout 1 ms.
8. The Test Station will transmit BREAK_REPLY primitive (6 times) as an invalid response. Then, it will send an OPEN request immediately after the primitives.

Observable Results: Verify that the DUT discards the OPEN request while waiting for BREAK response with timeout of 1 ms.
Figure 26 - Receive BREAK redundant primitive w/BREAK_REPLY Disabled
**SSP ERROR HANDLING – Enable and verify BREAK_REPY and Receive BREAK redundant primitive (TestD.7.2)**

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.

2. The Testing Station is instructed to transmit an Open Address frame to the DUT, followed by BREAK primitives (6 times).

Observable Results: Verify that the DUT transmits BREAK_REPLY primitive (6 times) response sequence within the 1 ms break timeout period.

3. The Testing Station is instructed to transmit an Open Address frame to the DUT. Test station will wait for DUT to accept Open request.

4. Wait for DUT to transmit BREAK primitive (6 times) due to connection idle timeout 1 ms.

5. The Test Station will transmit BREAK_REPLY primitive (6 times) as an invalid response.

Observable Results: Verify that the DUT does nothing

6. The Testing Station is instructed to transmit an Open Address frame to the DUT. Test station will wait for DUT to accept Open request.

7. Wait for DUT to transmit BREAK primitive (6 times) due to connection idle timeout 1 ms.

8. The Test Station will transmit BREAK primitive (6 times) as a collision BREAK condition.

Observable Results: Verify that the DUT transmits BREAK_REPLY primitives. After receiving the correct reply, the test station will transmit BREAK_REPLY primitive (6 times)
Figure 27 - Receive BREAK redundant primitive w/BREAK_REPLY Enabled
**BREAK TIMEOUT after sending BREAK redundant primitive (TestD.8)**

If the Target device sends a BREAK redundant primitive, it should receive a BREAK or BREAK_REPLY redundant primitive in response within 1 ms. If the Target device does not receive this response within 1 ms, it shall consider the connection broken.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT.
3. Wait for the DUT to transmit OPEN_ACCEPT and RRDY. The Testing Station is instructed to transmit a SCSI INQUIRY command to the DUT then close the connection.
4. Wait for the DUT to open an SSP connection to the Testing Station. Allow the DUT to transmit a SCSI Response to the Testing Station.
5. When the DUT transmits CLOSE (NORMAL) to close the connection, the Testing Station is instructed to not transmit CLOSE (NORMAL) in response.
6. Wait for the DUT to transmit BREAK primitive sequence (6 times). The Testing Station is instructed to not respond to the sequence.

Observable Results: Verify that the DUT after 1 ms timeout period does nothing and assumes the connection is closed.
Target assumes that the connection is closed and resumes normal operations.

Figure 28 - BREAK TIMEOUT after sending BREAK
**Receive BROADCAST(CHANGE) redundant primitive (TestD.9)**

SAS target ports shall ignore BROADCAST (CHANGE). This primitive is only used by Initiators and Expanders.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to transmit BROADCAST (CHANGE) Primitive to DUT.

Observable Results: Verify that the DUT discards the primitive with no additional action.

---

**Figure 29 - Broadcast discard**

![Diagram showing the interaction between Initiator/Test Station and Target Primitive Logic, Target Protocol Logic, including the broadcast primitive ignored.]
BREAK during Write or Read Sequence recovery (TestD.10.2 - TestD.10.3)
SAS target ports shall.

**SSP ERROR HANDLING – Break on Read Data frame sequence (TestD.10.2)**

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a READ(10) command with LBA=0 and 2 512 byte frames then close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit DATA frame.
4. The Testing Station is instructed to send BREAK (6 primitives) to the DUT. And, wait for BREAK or BREAK_REPLY response back from DUT.

Observable Results: Verify that the DUT responds to the write data sequence with a Response of CHECK CONDITION with ABORT COMMAND and ASC/Q of ACK/NAK TIMEOUT.
Figure 30 - Break on Read Data Recovery

Status = CHECK CONDITION, Sense Key = ABORT COMMAND
ASC/Q = ACK/NAK TIMEOUT
SSP ERROR HANDLING – Break on Write Data frame sequence (TestD.10.3)

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to transmit a WRITE10 command to the DUT and wait for the DUT to return and XFER_RDY response.
3. The Testing Station is then instructed to open an SSP connection to the DUT and to transmit a single DATA Frame.
4. Immediately after the EOF of the DATA frame, the Test Station is instructed to send 6 BREAK primitives.
5. The Test Station will wait for the BREAK or BREAK_REPLY primitive response. After receiving the response, The test station is instructed to send another 6 BREAK primitive sequence.

Observable Results: Verify that the DUT responds to the BREAK with another BREAK response.

6. If the DUT does not reply, then the Testing Station is instructed to open an SSP connection to the DUT.
7. Wait for the DUT to ACCEPT and DONE(NORMAL) the connection request, then initiate a connection closure with a DONE (ACK NAK TIMEOUT) primitive (Lost DATA Frame). The DUT and Test Station will initiate a CLOSE primitive sequence.

Observable Results: Verify that the DUT responds to the initial write request with a status of CHECK CONDITION with ASC/Q of INITIATOR RESPONSE TIMEOUT.
Figure 31 - Break on Write Data Frame and Recovery

Status = CHECK CONDITION, Sense Key = ABORT COMMAND
ASC/Q = INITIATOR RESPONSE
Error Conditions – Connection not Established (Test Group E)

Receive OPEN_REJECT (Retry class) (TestE.1)

Eighteen variants of the OPEN_REJECT primitive are defined in SAS2r14b.

Nine of these variants are in the Retry class. These represent temporary conditions in the SAS domain that should clear in a short time and allow the OPEN request to succeed if it is retried.

When the Target device receives a retry class OPEN_REJECT, it shall resend the OPEN_ADDRESS frame.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.
2. The Testing Station is instructed to open an SSP connection to the Testing Station.
3. Wait for the DUT to transmit OPEN_ACCEPT and RRDY. The Testing Station is instructed to transmit a SCSI INQUIRY command to the DUT then close the connection.
4. Wait for the DUT to transmit OPEN request for the response. The Testing Station is instructed to transmit a OPEN_REJECT (RETRY)

Observable Results: Verify that the DUT re-transmit the OPEN_ADDRESS request.
Figure 32 - Receive OPEN_REJECT(retry class)
Receive OPEN_REJECT (abandon class) (TestE.2)

The other nine variants of OPEN_REJECT are in the Abandon class. An abandon class OPEN_REJECT indicates a persistent condition that will cause OPEN_REJECT indefinitely if the OPEN is retried. An example is OPEN_REJECT (PROTOCOL NOT SUPPORTED)

When the Target device receives an abandon class OPEN_REJECT, it shall not resend the OPEN_ADDRESS frame. Receipt of an abandon class OPEN_REJECT shall be handled as an I_T Nexus Loss condition.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.
2. The Testing Station is instructed to open an SSP connection to the Testing Station.
3. Wait for the DUT to transmit OPEN_ACCEPT and RRDY. The Testing Station is instructed to transmit a SCSI INQUIRY command to the DUT then close the connection.
4. Wait for the DUT to transmit OPEN request for the response. The Testing Station is instructed to transmit a OPEN_REJECT (PROTOCOL NOT SUPPORTED)

Observable Results: Verify that the DUT does nothing before I_T Nexus loss period.
Figure 33 - Receive OPEN_REJECT (abandon class)
Receive OPEN_REJECT for unsupported protocol type (TestE.3)

SAS target ports shall reject and Open Request of an unsupported protocol type.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is an initiator.
2. The Testing Station is instructed to transmit an OpenAddress frame to the DUT with an STP type initiator role. The target does not support STP connection.
3. Wait for the DUT to transmit OPEN_REJECT (Protocol not supported).

Observable Results: Verify that the DUT responded to the Open Address frame with OPEN_REJECT (Protocol not supported).
Figure 34 - Reject Open of unsupported protocol type
Behavior After Receiving a Task Management Request (Test Group F)

ABORT TASK (TestF.1)

When ABORT TASK is received the Target device shall write any cached data to the media. The Target device will scan its Task Set for a command with this I_T_L_Q nexus. If such a command is present, it shall be aborted and no further frames associated with the command shall be sent by the Target device.

The Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE. This response code is used to indicate either that the identified command was aborted or that a command with that tag was not found in the task set.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI Test Unit Ready command to the DUT. This should not be the first command received by the DUT since power on. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit a SCSI response frame to the received command.
4. The Testing Station is instructed to open an SSP connection to the DUT and transmit a TASK MANAGEMENT frame ABORT_TASK. When ACK is completed, close the connection.
5. The Testing Station will allow the DUT to open an SSP connection.

Observable Results: Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE.

6. Repeat 1-5 using an invalid LUN (0xFFFFFFFF FFFFFFFF) address

Observable Results: Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID LUN.
Figure 35 - Abort Task
**ABORT TASK SET (TestF.2)**

When ABORT TASK SET is received the Target device shall write any cached data to the media. The Target device shall abort all commands in the current task set for this I_T_L nexus. Commands received on a different I_T_L nexus shall not be affected.

Pending status and sense data for the aborted commands shall be cleared. Other previously established conditions, such as mode parameters and reservations, shall not be changed.

The Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE.

**Test Procedure:**

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI Test Unit Ready command to the DUT. This should not be the first command received by the DUT since power on. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit a SCSI response frame to the received command.
4. The Testing Station is instructed to open an SSP connection to the DUT and transmit a TASK MANAGEMENT frame ABORT_TASK_SET. When ACK is completed, close the connection.
5. The Testing Station will allow the DUT to open an SSP connection.

**Observable Results:** Verify that the DUT transmit a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE.

6. Repeat 1-5 using an invalid LUN (0xFFFFFFFF FFFFFFFF) address

**Observable Results:** Verify that the DUT transmit a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID LUN.
Figure 36 - Abort Task Set
CLEAR ACA (TestF.3)

When CLEAR ACA is received, the Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 04h, TASK MANAGEMENT FUNCTION NOT SUPPORTED.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI Test Unit Ready command to the DUT. This should not be the first command received by the DUT since power on. Close the connection
3. Allow the DUT to open an SSP connection to the Testing Station and transmit a SCSI response frame to the received command.
4. The Testing Station is instructed to open an SSP connection to the DUT and transmits a TASK MANAGEMENT frame CLEAR_ACA. When ACK is completed, close the connection.
5. The Testing Station will allow the DUT to open an SSP connection.

Observable Results: Verify that the DUT transmit a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 04h, TASK MANAGEMENT FUNCTION NOT SUPPORTED.

6. Repeat 1-5 using an invalid LUN (0xFFFFFFFF FFFFFFFF) address

Observable Results: Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID LUN
Figure 37 - Clear ACA

Valid LUN = 0x00000000 00000000
SAS Protocol

DataPres = RESPONSE_DATA
Response_Code = TMF COMPLETE

Invalid LUN = FFFFFFFF FFFFFFFF
SAS Protocol

DataPres = RESPONSE_DATA
Response_Code = INVALID LUN
CLEAR TASK SET (TestF.4)

When CLEAR TASK SET is received, the Target device shall abort all commands in the current task set. SAS Target devices maintain a single task set for all initiators (Control mode page TST field = 000b, non-changeable), so this function aborts all commands pending in the Target device. Pending status and sense data for the aborted commands shall be cleared. Other previously established conditions, such as mode parameters and reservations, shall not be changed.

The Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI Test Unit Ready command to the DUT. This should not be the first command received by the DUT since power on. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit a SCSI response frame to the received command.
4. The Testing Station is instructed to open an SSP connection to the DUT and transmit a TASK MANAGEMENT frame CLEAR_TASK_SET. When ACK is completed, close the connection.
5. The Testing Station will allow the DUT to open an SSP connection.

Observable Results: Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE.

6. Repeat 1-5 using an invalid LUN (0xFFFFFFFF FFFFFFFF) address

Observable Results: Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID LUN
Figure 38 - Clear Task Set
**I_T NEXUS RESET (TestF.5)**

When I_T NEXUS RESET is received, the Target device shall
- abort all commands that have been received on the affected I_T nexus
- terminate all task management functions received on the affected I_T nexus
- establish a Unit Attention condition for the initiator associated with the affected I_T nexus

The Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE.

**Test Procedure:**

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI Test Unit Ready command to the DUT. This should not be the first command received by the DUT since power on. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit a SCSI response frame to the received command.
4. The Testing Station is instructed to open an SSP connection to the DUT and transmit a TASK MANAGEMENT frame I_T NEXUS_RESET. When ACK is completed, close the connection.
5. The Testing Station will allow the DUT to open an SSP connection.

**Observable Results:** Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE.

6. Repeat 1-5 using an invalid LUN (0xFFFFFFFF FFFFFFFF) address

**Observable Results:** Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID LUN.
Figure 39 - I_T NEXUS Open Timeout
LOGICAL UNIT RESET (TestF.6)

When LOGICAL UNIT RESET is received, the Target device shall
- abort all commands that have been received on the affected I_T nexus
- terminate all task management functions received on the affected I_T nexus
- establish a Unit Attention condition

The Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE.

When the Unit Attention condition resulting from this task management function is presented to an initiator, the ASC/ASCQ in that response frame shall be BUS DEVICE RESET FUNCTION OCCURRED.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI Test Unit Ready command to the DUT. This should not be the first command received by the DUT since power on. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit a SCSI response frame to the received command.
4. The Testing Station is instructed to open an SSP connection to the DUT and transmit a TASK MANAGEMENT frame LOGICAL_UNIT_RESET. When ACK is completed, close the connection.
5. The Testing Station will allow the DUT to open an SSP connection.

Observable Results: Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE.

6. Repeat 1-5 using an invalid LUN (0xFFFFFFFF FFFFFFFF) address

Observable Results: Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID LUN
**Figure 40 - Logical Unit Reset**

- **Task Mgmt Frame**
  - LOGICAL UNIT RESET
  - **ACK Primitive**
  - **SSP Link**
  - **SAS Protocol**
    - Response Frame
      - DataPres = RESPONSE_DATA
      - Response_Code = TMF COMPLETE
  - **Response Frame**
    - DataPres = RESPONSE_DATA
    - Response_Code = INVALID LUN

- **Task Mgmt Frame**
  - LOGICAL UNIT RESET
  - **ACK Primitive**
  - **SSP Link**
  - **SAS Protocol**
    - Response Frame
      - DataPres = RESPONSE_DATA
      - Response_Code = TMF COMPLETE
  - **Response Frame**
    - DataPres = RESPONSE_DATA
    - Response_Code = INVALID LUN

- **SSP Link**
  - **ACK Primitive**
  - **SAS Protocol**
    - Response Frame
      - DataPres = RESPONSE_DATA
      - Response_Code = TMF COMPLETE
  - **Response Frame**
    - DataPres = RESPONSE_DATA
    - Response_Code = INVALID LUN
QUERY TASK (TestF.7)

When QUERY TASK is received, the Target device shall determine whether the specified command is in the Target device’s task set.

If the specified command is present, then the Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 08h, TASK MANAGEMENT FUNCTION SUCCEEDED.

If the specified command is not present, then the Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI Test Unit Ready command to the DUT. This should not be the first command received by the DUT since power on. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit a SCSI response frame to the received command.
4. The Testing Station is instructed to open an SSP connection to the DUT and transmit a TASK MANAGEMENT frame QUERY_TASK. When ACK is completed, close the connection.
5. The Testing Station will allow the DUT to open an SSP connection.

Observable Results: Verify that the DUT transmit either a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE. Or, DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 08h, TASK MANAGEMENT FUNCTION SUCCEEDED.

6. Repeat 1-5 using an invalid LUN (0xFFFFFFFF FFFFFFFF) address

Observable Results: Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID LUN
Figure 41 - Query Task

- **Initiator/Test Station**
  - Task Mgmt Frame
  - QUERY TASK
  - ACK Primitive
  - Task Mgmt Frame
  - QUERY TASK
  - ACK Primitive

- **Target Primitive Logic**
  - SSP Link
  - ACK Primitive
  - SSP Link
  - ACK Primitive

- **Target Protocol Logic**
  - SAS Protocol
  - Response Frame
  - DataPres = RESPONSE_DATA
  - Response_Code = TMF COMPLETE
  - SSP Link
  - Valid LUN = 0x00000000 00000000
  - SAS Protocol
  - Response Frame
  - DataPres = RESPONSE_DATA
  - Response_Code = INVALID LUN
  - SSP Link
  - Invalid LUN = FFFFFFFF FFFFFFFF
  - SAS Protocol
  - Response Frame
  - DataPres = RESPONSE_DATA
  - Response_Code = INVALID LUN
QUERY TASK SET (TestF.8)

When QUERY TASK SET is received, the Target device shall determine whether there are any commands received for this I_T_L nexus in the Target device’s task set.

If any command is present, then the Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 08h, TASK MANAGEMENT FUNCTION SUCCEEDED.

If there are no commands present, then the Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI Test Unit Ready command to the DUT. This should not be the first command received by the DUT since power on. Close the connection
3. Allow the DUT to open an SSP connection to the Testing Station and transmit a SCSI response frame to the received command.
4. The Testing Station is instructed to open an SSP connection to the DUT and transmit a TASK MANAGEMENT frame QUERY_TASK_SET. When ACK is completed, close the connection.
5. The Testing Station will allow the DUT to open an SSP connection.

Observable Results: Verify that the DUT transmit either a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE. Or, DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 08h, TASK MANAGEMENT FUNCTION SUCCEEDED.

6. Repeat 1-5 using an invalid LUN (0xFFFFFFFF FFFFFFFF) address

Observable Results: Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID LUN
Figure 42 - Query Task Set
QUERY ASYNCHRONOUS EVENT (TestF.9)

When QUERY ASYNCHRONOUS EVENT is received, the Target device shall determine whether there is a Unit Attention condition pending for this I_T nexus.

If a Unit Attention is pending, then the Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 08h, TASK MANAGEMENT FUNCTION SUCCEEDED. It shall include Additional Response Information as defined in SAM4r14, 7.10.

If there is not a Unit Attention pending, then the Target device shall complete the task management request by sending a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE. It shall include Additional Response Information as defined in SAM4r14, 7.10.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI Test Unit Ready command to the DUT. This should not be the first command received by the DUT since power on. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit a SCSI response frame to the received command.
4. The Testing Station is instructed to open an SSP connection to the DUT and transmit a TASK MANAGEMENT frame QUERY_ASYNCHRONOUS_EVENT. When ACK is completed, close the connection.
5. The Testing Station will allow the DUT to open an SSP connection.

Observable Results: Verify that the DUT transmit either a Response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 00h, TASK MANAGEMENT FUNCTION COMPLETE. Or, DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to 08h, TASK MANAGEMENT FUNCTION SUCCEEDED.

6. Repeat 1-5 using an invalid LUN (0xFFFFFFFF FFFFFFFF) address

Observable Results: Verify that the DUT transmits a response frame with DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE field set to INVALID LUN
Figure 43 - Query Asynchronous Event
Task Management Function with Undefined and Reserved command (TestF.10 & TestF.11)

**SCSI CDB – Undefined Task Management Function (TestF.10)**

Verify that an undefined Task Management Function value has the proper response.

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 2048 bytes. Close the connection.
3. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY.
4. The Testing Station is then instructed to send an Undefined (0x03) Task Management function. The DUT should send an ACK response to the Command.

Observable Results: Verify that the DUT responds to the Undefined Task Management Function with a TASK MANAGEMENT NOT SUPPORTED

5. The Testing Station is instructed to open a connection to the DUT and transmit the 2048 bytes of data. Send 4 good DATA frames of 512 bytes each.

Observable Results: Verify that the DUT responds to the write data sequence with a Status=Good

**SCSI CDB – Reserved Task Management Function (TestF.11)**

Verify that a reserved Task Management Function value has the proper response.

Test Procedure:

   1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
   2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 2048 bytes. Close the connection.
   3. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY.
   4. The Testing Station is then instructed to send a Reserved (0x20) Task Management function. The DUT should send an ACK response to the Command.

Observable Results: Verify that the DUT responds to the Undefined Task Management Function with a TASK MANAGEMENT NOT SUPPORTED

5. The Testing Station is instructed to open a connection to the DUT and transmit the 2048 bytes of data. Send 4 good DATA frames of 512 bytes each.

Observable Results: Verify that the DUT responds to the Write data sequence with a Status=Good
Figure 44 - Task Mgmt with Undefined or Reserved Function
TMF or Commands w/Tag conflict during Read or Write command sequence (TestF.15.2 – TestF.15.5)

Verify that an undefined Task Management Function value has the proper response.

SCSI CDB – Identical Write command sequence with TAG Conflict (no retran bit) (TestF.15.2)

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.
2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 512 bytes w/Tag = 0018. Close the connection. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY w/Tag = 0018.
3. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 512 bytes w/Tag = 0019. Close the connection.
4. Before DUT responds, the Testing Station is instructed to open an SSP connection to the DUT and transmit an identical SCSI WRITE command as first Write w/Tag = 0018. Close the connection.

Observable Results: Verify that the DUT responds to the duplicate Write with a response of Status CHECK CONDITION, Sense Key of ABORT COMMAND and ASC/Q of OVERLAPPING COMMANDS ATTEMPTED.

5. Allow the DUT to open an SSP connection to the Testing Station and transmit XFER_RDY w/Tag = 0019.
6. The Testing Station is instructed to open a connection to the DUT and transmit the 512 byte data frame w/Tag = 0019.

Observable Results: Verify that the DUT responds to the second write with a response of Status GOOD from TAG = 0019.

7. The Testing Station is instructed to open a connection to the DUT and transmit a Test Unit Ready command frame w/Tag = 001A.

Observable Results: Verify that the DUT responds to the TUR data sequence with a Status=Good w/Tag = 001A
Figure 45 - Identical Write command sequence w/TAG Conflict
**SCSI CDB – Identical Read command sequence with TAG Conflict (no retran bit) (TestF.15.3)**

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.

2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI READ command for 512 bytes w/TAG = 0018. Close the connection.

3. Before DUT responds, the Testing Station is instructed to open an SSP connection to the DUT and transmit an identical SCSI READ command as first Write w/TAG = 0018. Close the connection.

Observable Results: Verify that the DUT responds to the duplicate write with a response of Status CHECK CONDITION, Sense Key of ABORT COMMAND and ASC/Q of OVERLAPPING COMMANDS ATTEMPTED.

4. The Testing Station is instructed to open a connection to the DUT and transmit a Test Unit Ready command frame w/TAG = 001A.

Observable Results: Verify that the DUT responds to the TUR data sequence with a Status=Good w/TAG = 001A
Figure 46 - Identical Read command sequence w/TAG Conflict (no retrans bit)
**SCSI CDB – Read and ABORT TASK command sequence with TAG Conflict (TestF.15.4)**

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.

2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI READ command for 512 bytes w/TAG = 0018. Close the connection.

3. Before DUT responds, the Testing Station is instructed to open an SSP connection to the DUT and transmit an ABORT TASK command w/TAG = 0018. Close the connection.

Observable Results: Verify that the DUT responds to the duplicate write with a response of Status CHECK CONDITION, Sense Key of ABORT COMMAND and ASC/Q of OVERLAPPING COMMANDS ATTEMPTED.

4. The Testing Station is instructed to open a connection to the DUT and transmit a Test Unit Ready command frame w/TAG = 0019.

Observable Results: Verify that the DUT responds to the TUR data sequence with a Status=Good w/TAG = 0019.
Figure 47 - Read and ABORT TASK command sequence w/TAG Conflict
SCSI CDB – Write and Query TASK command sequence with same TAG Conflict (TestF.15.5)

Test Procedure:

1. The Testing Station is instructed to start and complete a phy Reset sequence followed by an Identify sequence with the DUT. Since the DUT is a target, the Testing Station is instructed to transmit an Identify Address frame indicating that it is a SAS Initiator.

2. The Testing Station is instructed to open an SSP connection to the DUT and transmit a SCSI WRITE command for 512 bytes w/TAG = 0018. Close the connection.

3. Before DUT responds, the Testing Station is instructed to open an SSP connection to the DUT and transmit a QUERY TASK command w/TAG = 0018. Close the connection.

Observable Results: Verify that the DUT responds to the duplicate write with a response of Status CHECK CONDITION, Sense Key of ABORT COMMAND and ASC/Q of OVERLAPPING COMMANDS ATTEMPTED.

4. The Testing Station is instructed to open a connection to the DUT and transmit an ABORT TASK command w/TAG = 0019. Close the connection.

Observable Results: Verify that the DUT responds to the ABORT TASK sequence with a Status TMF Completed w/TAG = 0019.
Figure 48 - Write and Query TASK command sequence w/TAG Conflict