JEDEC STANDARD

Universal Flash Storage (UFS) Test Version 1.1

JESD224A

(Revision of JESD224, March 2013)

JULY 2017

JEDEC SOLID STATE TECHNOLOGY ASSOCIATION

NOTICE

JEDEC standards and publications contain material that has been prepared, reviewed, and approved through the JEDEC Board of Directors level and subsequently reviewed and approved by the JEDEC legal counsel.

JEDEC standards and publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for use by those other than JEDEC members, whether the standard is to be used either domestically or internationally.

JEDEC standards and publications are adopted without regard to whether or not their adoption may involve patents or articles, materials, or processes. By such action JEDEC does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the JEDEC standards or publications.

The information included in JEDEC standards and publications represents a sound approach to product specification and application, principally from the solid state device manufacturer viewpoint. Within the JEDEC organization there are procedures whereby a JEDEC standard or publication may be further processed and ultimately become an ANSI standard.

No claims to be in conformance with this standard may be made unless all requirements stated in the standard are met.

Inquiries, comments, and suggestions relative to the content of this JEDEC standard or publication should be addressed to JEDEC at the address below, or refer to www.jedec.org under Standards and Documents for alternative contact information.

Published by
©JEDEC Solid State Technology Association 2017
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

PRICE: Contact JEDEC

Printed in the U.S.A.

Universal Flash Storage (UFS) Test

Contents

Tables

Universal Flash Storage (UFS) Test

(From JEDEC Board Ballot JCB-13-18 and JCB-17-22, formulated under the cognizance of the JC-64.5 Subcommittee on UFS Measurement.)

1 Scope

The primary objective of this test standard is to specify the test cases for UFS device protocol conformance testing. This test standard provides test cases for checking the functions defined in the following target standard:

JESD220, Universal Flash Storage (UFS) Standard version 1.1A

MIPI M-PHY and MIPI UniPro test cases are not in the scope of this document.

2 Normative Reference

[UFS] JESD220C, Universal Flash Storage (UFS) version 2.1
[UFS Card] JESD220-2, Universal Flash Storage (UFS) Card Extension version 1.0
[UFSHCI] JESD223C, Universal Flash Storage (UFS) Host Controller Interface version 2.1
[SAM] SCSI Architecture Model-5(SAM-5) Revision05, 19 May 2010
[SPC] T10 Specification: SCSI Primary Commands – 4(SPC- 4) Revision 27, 11 October 2010
[SBC] T10 Specification: SCSI Block Commands – 3(SBC -3) Revision 24, 05 August 2010

3 Terms, Definitions, Acronyms, and Symbols

may: Indicates flexibility of choice with no implied recommendation or requirement.

shall: Indicates a mandatory requirement. Designers shall implement such mandatory requirements to ensure interchangeability and to claim conformance with the specification.

should: Indicates a strong recommendation but not a mandatory requirement. Designers should give strong consideration to such recommendations, but there is still a choice in implementation.

3.1 Acronyms

ATTRVALUE	Attribute Value
GROUPNUM	Group Number
LBA	Logical Block Address
LUN	Logical Unit Number
MIPI	Mobile Industry Processor Interface
PCODE	Page Code
RPMB	Replay Protected Memory Block
SPCODE	Sub Page Code
TSF	Transaction Specific Fields
UFS	Universal Flash Storage
UPIU	UFS Protocol Information Unit
UniPro	Unified Protocol
ASYNC	Out of order execution support

3.2 Naming Conventions

Some terms are capitalized to distinguish their definition from their common English meaning. Words not capitalized retain their common English meaning.

3.3 Numbers and Number Bases

Hexadecimal numbers are written with a lower case "h" suffix, e.g., FFFFh and 80h.

Binary numbers are written with a lower case "b" suffix (e.g., 10b).

Binary numbers larger than four digits are written with a space dividing each group of four digits, as in 1000 0101 0010b.

All other numbers are decimal.

3.4 Symbols

Dash (-) not applicable (n/a)

4 Introduction

Universal Flash Storage (UFS) is a simple, high performance, mass storage device with a serial Interface. It is primarily for use in mobile systems, between host processing and mass storage memory devices. The following is the summary of the UFS functional features,

- Similar functional features as eMMC
- Boot Operation Mode
- Device enumeration & discovery
- Supports Multiple partitions (LUNs) with partition Management
- Supports Multiple User Data Partition with Enhanced User Data Area options
- Support for boot partitions and RPMB partition
- Reliable write operation
- Background operations
- Secure operations, Purge and Erase to enhance data security
- Write Protection options, including Permanent & Power-On Write Protection
- Signed access to a Replay Protected Memory Block
- HW Reset Signals
- Task management operations
- Power management operations

4.1 Test specification Guiding Principles

1) Consistency – The principal is based on the need to generate a consistent result for all test items regardless of the testing implementation. The goal of this principal is to prevent a situation in which test implementations of a test item have different results due to a wide definition of the test item. According to this principal each test item provides detailed description for: commands, command sequence, specific command input parameters and specific expected output. Specific values are assigned to parameters, ranges are not used.

5 Test Case Specifications Format

The test case specification contains following fields,

- **Ref. specs Section**

  This field indicates the corresponding sections in the referenced document.

- **Test Purpose**

  This field briefs about the intention of the test case.

- **Test Procedure**

  This field contains test case execution procedure to be followed,

  [Precondition]

  The steps mentioned under this block shall be executed before main test case execution. The test steps mentioned in the **[Main]** block has dependency on these conditions. The output of test case may vary if precondition is not met.

  [Main]

  The test case execution procedure is mentioned in this section.

  [Clean up]

  Wherever mentioned, these steps to be followed after the test case execution to bring back the test environment to normal condition. These clean-up steps are needed to reset the test environment.

- **Input parameter values**

  The applicable values for the parameters used in the test case.

- **Expected Output**

  This field contains the expected result of the test case execution.

6 Assumption

- **Expected Data Transfer Length**
    - In this specification, all test cases assume that Expected Data Transfer Length value in UPIU equals to Allocation Length or Transfer Length in byte.
    - If Transfer Length equal to zero in Read6 and Write6 command, Data Transfer Length value is set to 256 Logical blocks in byte.

- **Device Specific values shall not be checked.**
- **Expected Output defines output which shall be tested as pass/fail criteria.**
- **Device Initialization and Configuration shall be completed and LU shall be ready.**

7 UFS SCSI Command Test

7.1 INQUIRY Command

The following test cases may be applied to all configured logical units and well-know logical units.
Standard INQUIRY data is 36bytes for UFS v1.1

7.1.1 Test Case Id: UFS_Inquiry_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that the INQUIRY command returns 36 bytes of Standard INQUIRY data.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]
None.

[Main]
1. Issue INQUIRY command.
2. Verify expected output.

[Clean up]
None.</td></tr>
<tr><td>Input parameter values</td><td>INQUIRY
EVPD = 0, PAGE CODE = 0, ALLOCATION LENGTH = 36, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The device shall return the 36 bytes of Standard INQUIRY data.</td></tr>
</table>

7.1.2 Test Case Id: UFS_Inquiry_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.2</td></tr>
<tr><td>Test Purpose</td><td>To verify status of the INQUIRY command with invalid page code parameter</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]
None

[Main]
1. Issue INQUIRY command.
2. Verify expected output.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>INQUIRY
EVPD = 0, PAGE CODE = 83h, ALLOCATION LENGTH = 36, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN CDB
or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.1.3 Test Case Id: UFS_Inquiry_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that the INQUIRY command returns 36 bytes of Standard INQUIRY data if the ALLOCATION LENGTH is greater than 36</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]
None

[Main]
1. Issue INQUIRY command.
2. Verify expected output.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>INQUIRY
EVPD = 0, PAGE CODE = 00h, ALLOCATION LENGTH = 37, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. INQUIRY command shall return 36 bytes of Standard INQUIRY data
3. Underflow flag in RESPONSE UPIU shall be set to one.</td></tr>
</table>

7.1.4 Test Case Id: UFS_Inquiry_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that the INQUIRY command returns the requested number of data bytes when the ALLOCATION LENGTH is less than 36.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]
None.

[Main]
1. Issue INQUIRY command.
2. Verify expected output.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>INQUIRY
EVPD = 0, PAGE CODE = 00h, ALLOCATION LENGTH = 35, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. INQUIRY command shall return the amount of data specified in the ALLOCATION LENGTH field (i.e. 35 bytes).</td></tr>
</table>

7.1.5 Test Case Id: UFS_Inquiry_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.2</td></tr>
<tr><td>Test Purpose</td><td>To verify the INQUIRY command with pending UNIT ATTENTION condition.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]
1. Execute hardware reset to establish UNIT ATTENTION condition and not send REQUEST SENSE command.
[Main]
1. Issue INQUIRY command
2. Verify expected output.
[Clean up]
1. Send REQUEST SENSE command to clear the UNIT ATTENTION condition.</td></tr>
<tr><td>Input parameter values</td><td>INQUIRY
EVPD = 0, Page code = 00h, Allocation length = 36, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. INQUIRY command shall return 36 bytes of standard inquiry data</td></tr>
</table>

7.2 REQUEST SENSE Command

The following test cases may be applied to all configured logical units and well-know logical units.

Sense Data is 18 bytes for UFS.

7.2.1 Test Case Id: UFS_RequestSense_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.17</td></tr>
<tr><td>Test Purpose</td><td>To verify the REQUEST SENSE command with DESC = 0 (fixed format sense data)</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue REQUEST SENSE command.
2. Verify expected output.

[Clean up]:
None.</td></tr>
<tr><td>Input parameter values</td><td>REQUEST SENSE
DESC = 0, Allocation Length = 12h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The value of the following fields shall be verified: Response Code=70h, ADDITIONAL SENSE LENGTH=0Ah</td></tr>
</table>

7.2.2 Test Case Id: UFS_RequestSense_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.17</td></tr>
<tr><td>Test Purpose</td><td>To verify that after receiving auto sense data for a command terminated with CHECK CONDITION status, a subsequent REQUEST SENSE command returns GOOD status and sense key in parameter data set to NO SENSE.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue MODE SENSE command with InvalidPageCode[1)] value.
2. Verify that MODE SENSE command is terminated with CHECK CONDITION status.
3. Verify Auto Sense Data (Sense key set to ILLEGAL REQUEST).
[Main]
1. Issue REQUEST SENSE command.
2. Verify expected output.
[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>REQUEST SENSE
DESC = 0, ALLOCATION LENGTH = 12h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. REQUEST SENSE command shall return sense data with the sense key set to NO SENSE (00h) and the additional sense code set to NO ADDITIONAL SENSE INFORMATION (ASC = 00h, ASCQ = 00h).</td></tr>
</table>

NOTE 1 InvalidPageCode value shall be equal to the minimum Page Code value (except zero) not supported by the device. For example, if a device supports only the values:

01h (READ-WRITE ERROR RECOVERY mode page),

08h (CACHING mode page),

0Ah (CONTROL mode page),

the InvalidPageCode shall be set to 02h.

The list of supported mode pages may be obtained issuing MODE SENSE (10) command with Page Code set to 3F (ALL PAGES).

7.2.3 Test Case Id: UFS_RequestSense_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.17</td></tr>
<tr><td>Test Purpose</td><td>To verify that the REQUEST SENSE command returns 18 data bytes if the allocation length is greater than 18</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue REQUEST SENSE command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>REQUEST SENSE
DESC = 0, ALLOCATION LENGTH = 13h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The REQUEST SENSE command shall return 18 bytes of data
3. The value of the following fields shall be verified: RESPONSE CODE=70h, ADDITIONAL SENSE LENGTH=0Ah.
4. The Underflow flag in RESPONSE UPIU shall be set to one.</td></tr>
</table>

7.2.4 Test Case Id: UFS_RequestSense_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.17</td></tr>
<tr><td>Test Purpose</td><td>To verify that the REQUEST SENSE command returns the requested number of data bytes if the allocation length is less than 18</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue REQUEST SENSE command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>REQUEST SENSE
DESC = 0, ALLOCATION LENGTH = 11h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The REQUEST SENSE command shall return 17 bytes of data
3. The value of the following fields shall be verified: RESPONSE CODE=70h, ADDITIONAL SENSE LENGTH=0Ah.</td></tr>
</table>

7.3 MODE SENSE Command

The following test cases shall be applied to all configured logical units.

7.3.1 Test Case Id: UFS_ModeSense_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify status of MODE SENSE (10) command with the Allocation Length = 0</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 00b, 01b, 10b; LLBAA = 0, DBD = 1, PAGE CODE = Ah,
SUBPAGE CODE = 00h, ALLOCATION LENGTH = 0h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. No data shall be returned.</td></tr>
</table>

7.3.2 Test Case Id: UFS_ModeSense_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 0Ah (Control mode) and PC = 00b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1, PAGE CODE = 0Ah, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 14h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Control mode page content is device specific.</td></tr>
</table>

7.3.3 Test Case Id: UFS_ModeSense_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 0Ah (Control mode) and PC = 01b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 01b, LLBAA = 0, DBD = 1, PAGE CODE = 0Ah, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 14h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The value of the following fields shall be verified: TST=000b, SWP=1b and 0000h</td></tr>
</table>

7.3.4 Test Case Id: UFS_ModeSense_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 0Ah (Control mode) and PC = 10b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None.</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 10b, LLBAA = 0, DBD = 1,PAGE CODE = 0Ah, SUBPAGE CODE = 00h,
ALLOCATION LENGTH = 14h CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Control mode page content is device specific.</td></tr>
</table>

7.3.5 Test Case Id: UFS_ModeSense_05

This test case may be applied only if PS bit of the Control mode page is equal to one.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 0Ah (Control mode) and PC = 11b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1, PAGE CODE = 0Ah, SUBPAGE CODE = 00h,
ALLOCATION LENGTH = 14h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Control mode page content is device specific.</td></tr>
</table>

7.3.6 Test Case Id: UFS_ModeSense_06

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 01h (Read-Write Error recovery mode) and PC = 00b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 14h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Read-Write Error recovery mode page content is device specific.</td></tr>
</table>

7.3.7 Test Case Id: UFS_ModeSense_07

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 01h (Read-Write Error recovery mode) and PC = 01b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 01b, LLBAA = 0, DBD = 1, PAGE CODE = 01h, SUBPAGE CODE = 00h, ALLOCATION LENGTH, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
</td></tr>
</table>

7.3.8 Test Case Id: UFS_ModeSense_08

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 01h (Read-Write Error recovery mode) and PC = 10b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 10b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 14h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Read-Write Error recovery mode page content is device specific.</td></tr>
</table>

7.3.9 Test Case Id: UFS_ModeSense_09

This test case may be applied only if PS bit of the Read-Write Error recovery mode page is equal to one.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 01h (Read-Write Error recovery mode) and PC = 11b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1, PAGE CODE = 01h, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 14h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Read-Write Error recovery mode page content is device specific.</td></tr>
</table>

7.3.10 Test Case Id: UFS_ModeSense_10

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 08h (caching mode) and PC = 00b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 08h, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 1Ch, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Caching mode page content is device specific</td></tr>
</table>

7.3.11 Test Case Id: UFS_ModeSense_11

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 08h (caching mode) and PC = 01b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 01b, LLBAA = 0, DBD = 1, PAGE CODE = 08h, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 1Ch, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Only the value of the following fields shall be verified: WCE= 1b, RCD = 1b</td></tr>
</table>

7.3.12 Test Case Id: UFS_ModeSense_12

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 08h (caching mode) and PC = 10b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 10b, LLBAA = 0, DBD = 1,PAGE CODE = 08h, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 1Ch, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Caching mode page content is device specific.</td></tr>
</table>

7.3.13 Test Case Id: UFS_ModeSense_13

This test case may be applied only if PS bit of the Caching mode page is equal to one.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 08h (caching mode) and PC = 11b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1, PAGE CODE = 08h, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 1Ch, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The returned page data contains the details of saved values of Control mode page.</td></tr>
</table>

7.3.14 Test Case Id: UFS_ModeSense_14

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with
PAGE CODE = 3Fh (All mode pages) and PC = 00b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1, PAGE CODE = 3Fh, SUBPAGE CODE = 00h, ALLOCATION LENGTH = FFFFh, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. See UFS_ModeSense_02 for Control Mode Page content, UFS_ModeSense_06 for Read-Write Error recovery Mode Page content and UFS_ModeSense_10 for caching mode page content
NOTE The data shall be in ascending order of Mode Page code.</td></tr>
</table>

7.3.15 Test Case Id: UFS_ModeSense_15

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SENSE (10) command with invalid page code parameter.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 00b,01b,10b,11b; LLBAA = 0, DBD = 1, PAGE CODE = InvalidPageCode(1), SUBPAGE CODE = 00h, ALLOCATION LENGTH = 1Ch, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN CDB or NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

NOTE 1 InvalidPageCode value shall be equal to the minimum Page Code value (except zero) not supported by the device. For example, if a device supports only the values:

01h (READ-WRITE ERROR RECOVERY mode page),

08h (CACHING mode page),

0Ah (CONTROL mode page),

the InvalidPageCode shall be set to 02h.

The list of supported mode pages may be obtained issuing MODE SENSE (10) command with Page Code set to 3F (ALL PAGES).

7.3.16 Test Case Id: UFS_ModeSense_16

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the MODE SENSE (10) command status with ALLOCATION LENGTH value lower than the Control Mode page size.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1, PAGE CODE = 0Ah, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 13h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The device server shall transfer a number of bytes equal to the ALLOCATION LENGTH value (i.e 19 bytes).
3. See UFS_ModeSense_02 for Control Mode Page content.</td></tr>
</table>

7.3.17 Test Case Id: UFS_ModeSense_17

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify that the MODE SENSE (10) command returns only the Control mode page content if ALLOCATION LENGTH value is greater than 20.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 10b, LLBAA = 0, DBD = 1, PAGE CODE = 0Ah, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 15h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The device shall transfer only 20 bytes equal and not the amount bytes specified in ALLOCATION LENGTH field.
3. See UFS_ModeSense_02 for Control Mode Page content.
4. Underflow flag in RESPONSE UPIU shall be set to one.</td></tr>
</table>

7.3.18 Test Case Id: UFS_ModeSense_18

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify that the Mode parameter header values.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1, PAGE CODE = 0Ah, SUBPAGE CODE = 00h, ALLOCATION LENGTH = 14h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The value of the following fields shall be verified: MODE DATA LENGTH = 18.</td></tr>
</table>

7.3.19 Test Case Id: UFS_ModeSense_19

This test case shall be applied only if the device does not support the Control Extension mode page.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify that MODE SENSE (10) command is terminated with CHECK CONDITION status when a Subpage is sensed and the device does not support the Subpage.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]

[Main]
1. Issue MODE SENSE (10) command.
2. Verify expected output.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1, PAGE CODE = 0Ah,
SUBPAGE CODE = 01h, ALLOCATION LENGTH = 28h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN CDB or NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.4 MODE SELECT (10) Command

The following test cases may be applied to all configured logical units.

7.4.1 Test Case Id: UFS_ModeSelect10_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify that the MODE SELECT (10) command changes the current value of SWP parameter in Control mode page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:

[Main]:
1. Issue MODE SENSE(10) command for getting SWP(N1) and BUSY TIMEOUT PERIOD(N2) current value.<a>
2. Issue MODE SELECT (10) command.
3. Verify expected output.<1>
4. Issue MODE SENSE (10) command. <c>
5. Verify expected output.<2>

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 0Ah, Subpage code = 00h, ALLOCATION LENGTH = 14h

b) MODE SELECT (10)
PF = 1, SP = 0, Parameter List Length = 14h
SWP = N(N != N1) and BUSY TIMEOUT PERIOD = N2 and all other fields set to zero.

c) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 0Ah, Subpage code = 00h, ALLOCATION LENGTH = 14h</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: SWP=N and BUSY TIMEOUT PERIOD = N2 and all other fields are zero.</td></tr>
</table>

7.4.2 Test Case Id: UFS_ModeSelect10_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify that the MODE SELECT (10) command changes the current value of BUSY TIME PERIOD parameter in Control mode page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:

[Main]:
1. Issue MODE SENSE(10) command for getting SWP(N1) and BUSY TIMEOUT PERIOD(N2) current value.<a>
2. Issue MODE SELECT (10) command. <a>
3. Verify expected output.<1>
4. Issue MODE SENSE (10) command.
5. Verify expected output.<2>

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 0Ah, Subpage code = 00h, ALLOCATION LENGTH = 14h

b) MODE SELECT (10)
PF = 1, SP = 0, Parameter List Length = 14h
SWP = N1, BUSY TIME PERIOD = N(N != N2) and all other fields set to zero.

c) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 0Ah, Subpage code = 00h, ALLOCATION LENGTH = 14h</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: SWP = N1 and BUSY TIMEOUT PERIOD = N and all other fields are zero.</td></tr>
</table>

7.4.3 Test Case Id: UFS_ModeSelect10_03

This test case may be applied when READ RETRY COUNT parameter is changeable.

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of the MODE SELECT (10) command changes the current value of READ RETRY COUNT parameter in Read-Write Error Recovery Mode Page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE(10) command for getting READ RETRY COUNT(N1), WRITE RETRY COUNT(N2)and RECOVERY TIME LIMIT(N3) current value.<a>
2. Issue MODE SELECT (10) command.
3. Verify expected output <1>
4. Issue MODE SENSE (10) command. <c>
5. Verify expected output<2>.

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h

b) MODE SELECT (10)
PF = 1, SP = 0, Parameter List Length = 14h
AWRE = 1b, READ RETRY COUNT = N(N != N1), WRITE RETRY COUNT = N2, RECOVERY TIME LIMIT = N3 and all other fields set to zero.

c) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: AWRE = 1b, READ RETRY COUNT= N, WRITE RETRY COUNT = N2, RECOVERY TIME LIMIT = N3 and all other fields are zero.</td></tr>
</table>

7.4.4 Test Case Id: UFS_ModeSelect10_04

This test case may be applied when WRITE RETRY COUNT parameter is changeable.

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of the MODE SELECT (10) command changes the current value of WRITE RETRY COUNT parameter in Read-Write Error Recovery Mode Page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE(10) command for getting READ RETRY COUNT(N1), WRITE RETRY COUNT(N2) and RECOVERY TIME LIMIT(N3) current value.<a>
2. Issue MODE SELECT (10) command.
3. Verify expected output <1>
4. Issue MODE SENSE (10) command. <c>
5. Verify expected output<2>.

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h

b) MODE SELECT (10)
PF = 1, SP = 0, Parameter List Length = 14h
AWRE = 1b, READ RETRY COUNT = N1, WRITE RETRY COUNT = N(N != N2), RECOVERY TIME LIMIT = N3 and all other fields set to zero.

c) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: AWRE = 1b, READ RETRY COUNT = N1, WRITE RETRY COUNT = N, RECOVERY TIME LIMIT = N3 and all other fields are zero.</td></tr>
</table>

7.4.5 Test Case Id: UFS_ModeSelect10_05

This test case may be applied when RECOVERY TIME LIMIT parameter is changeable.

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of the MODE SELECT (10) command changes the current value of RECOVERY TIME LIMIT parameter in Read-Write Error Recovery Mode Page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE(10) command for getting READ RETRY COUNT(N1), WRITE RETRY COUNT(N2) and RECOVERY TIME LIMIT(N3) current value.<a>
2. Issue MODE SELECT (10) command.
3. Verify expected output <1>
4. Issue MODE SENSE (10) command. <c>
5. Verify expected output<2>.

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h

b) MODE SELECT (10)
PF = 1, SP = 0, Parameter List Length = 14h
AWRE = 1b, READ RETRY COUNT = N1, WRITE RETRY COUNT = N2, RECOVERY TIME LIMIT = N(N != N3) and all other fields set to zero.

c) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: AWRE = 1b, READ RETRY COUNT = N1, WRITE RETRY COUNT = N2, RECOVERY TIME LIMIT = N and all other fields are zero.</td></tr>
</table>

7.4.6 Test Case Id: UFS_ModeSelect10_06

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of the MODE SELECT (10) command changes the current value of WCE parameter in Caching Mode Page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:

[Main]:
1. Issue MODE SENSE(10) command for getting WCE(N1) and RCD(N2) current value.<a>
2. Issue MODE SELECT (10) command. <a>
3. Verify expected output <1>
4. Issue MODE SENSE (10) command.
5. Verify expected output<2>.

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 08h, Subpage code = 00h, ALLOCATION LENGTH = 1Ch

b) MODE SELECT (10)
PF = 1, SP = 0, Parameter List Length = 1Ch
WCE = N(N != N1), RCD = N2 and all other fields set to zero.

c) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 08h, Subpage code = 00h, ALLOCATION LENGTH = 1Ch</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: WCE = N, RCD = N2, all other fields are zero.</td></tr>
</table>

7.4.7 Test Case Id: UFS_ModeSelect10_07

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of the MODE SELECT (10) command changes the current value of RCD parameter in Caching Mode Page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:

[Main]:
1. Issue MODE SENSE(10) command for getting WCE(N1) and RCD(N2) current value.<a>
2. Issue MODE SELECT (10) command. <a>
3. Verify expected output<1>.
4. Issue MODE SENSE (10) command.
5. Verify expected output<2>.

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 08h, Subpage code = 00h, ALLOCATION LENGTH = 1Ch

a) MODE SELECT (10)
PF = 1, SP = 0, Parameter List Length = 1Ch
WCE = N1, RCD = N(N != N2) and all other fields set to zero.

b) MODE SENSE (10)
PC = 00b, LLBAA = 0, DBD = 1,PAGE CODE = 08h, Subpage code = 00h, ALLOCATION LENGTH = 1Ch</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: WCE = N1, RCD = N and all other fields are zero.</td></tr>
</table>

7.4.8 Test Case Id: UFS_ModeSelect10_08

This test case may be applied only if the Control mode page is savable.

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify that the MODE SELECT (10) command changes the saved value of SWP parameter in Control mode page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE(10) command for getting SWP(N1) and BUSY TIMEOUT PERIOD(N2) saved value.<a>
2. Issue MODE SELECT (10) command.
3. Verify expected output.<1>
4. Reset the device(Power-on reset, H/W reset, End point reset and Host system UniPro reset)
5. Issue MODE SENSE (10) command. <c>
6. Verify expected output.<2>

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 0Ah, Subpage code = 00h, ALLOCATION LENGTH = 14h

b) MODE SELECT (10)
PF = 1, SP = 1, Parameter List Length = 14h
SWP = N(N != N1) and BUSY TIMEOUT PERIOD = N2 and all other fields set to zero.

c) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 0Ah, Subpage code = 00h, ALLOCATION LENGTH = 14h</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: SWP = N and BUSY TIMEOUT PERIOD = N2 and all other fields are zero.</td></tr>
</table>

7.4.9 Test Case Id: UFS_ModeSelect10_09

This test case may be applied only if the Control mode page is savable..

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify that the MODE SELECT (10) command changes the saved value of BUSY TIME PERIOD parameter in Control mode page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE(10) command for getting SWP(N1), BUSY TIMEOUT PERIOD(N2) saved value.<a>
2. Issue MODE SELECT (10) command.
3. Verify expected output.<1>
4. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset)
5. Issue MODE SENSE (10) command.<c>
6. Verify expected output.<2>

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 0Ah, Subpage code = 00h, ALLOCATION LENGTH = 14h

b) MODE SELECT (10)
PF = 1, SP = 1, Parameter List Length = 14h
SWP = N1, BUSY TIME PERIOD = N(N != N2) and all other fields set to zero.

c) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 0Ah, Subpage code = 00h, ALLOCATION LENGTH = 14h</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: SWP = N1 and BUSY TIMEOUT PERIOD = N, all other fields are zero.</td></tr>
</table>

7.4.10 Test Case Id: UFS_ModeSelect10_10

This test case may be applied only if the Read-Write Error Recovery mode page is savable and READ RETRY COUNT parameter is changeable.

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of the MODE SELECT (10) command changes the saved value of READ RETRY COUNT parameter in Read-Write Error Recovery Mode Page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE(10) command for getting READ RETRY COUNT(N1), WRITE RETRY COUNT(N2)and RECOVERY TIME LIMIT(N3) saved value.<a>
2. Issue MODE SELECT (10) command.
3. Verify expected output <1>
4. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset)
5. Issue MODE SENSE (10) command. <c>
6. Verify expected output<2>.

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h

b) MODE SELECT (10)
PF = 1, SP = 1, Parameter List Length = 14h
AWRE = 1b, READ RETRY COUNT = N(N != N1), WRITE RETRY COUNT = N2, RECOVERY TIME LIMIT = N3 and all other fields set to zero.

c) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: AWRE = 1b, READ RETRY COUNT= N, WRITE RETRY COUNT = N2, RECOVERY TIME LIMIT = N3 and all other fields are zero.</td></tr>
</table>

7.4.11 Test Case Id: UFS_ModeSelect10_11

This test case may be applied only if the Read-Write Error Recovery mode page is savable and WRITE RETRY COUNT parameter is changeable.

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of the MODE SELECT (10) command changes the saved value of WRITE RETRY COUNT parameter in Read-Write Error Recovery Mode Page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE(10) command for getting READ RETRY COUNT(N1), WRITE RETRY COUNT(N2) and RECOVERY TIME LIMIT(N3) saved value.<a>
2. Issue MODE SELECT (10) command.
3. Verify expected output <1>
4. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset)
5. Issue MODE SENSE (10) command. <c>
6. Verify expected output<2>.

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h

b) MODE SELECT (10)
PF = 1, SP = 1, Parameter List Length = 14h
AWRE = 1b, READ RETRY COUNT = N1, WRITE RETRY COUNT = N(N != N2), RECOVERY TIME LIMIT = N3 and all other fields set to zero.

c) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: AWRE = 1b, READ RETRY COUNT = N1, WRITE RETRY COUNT = N, RECOVERY TIME LIMIT = N3 and all other fields are zero.</td></tr>
</table>

7.4.12 Test Case Id: UFS_ModeSelect10_12

This test case may be applied only if the Read-Write Error Recovery mode page is savable and RECOVERY TIME LIMIT parameter is changeable.

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of the MODE SELECT (10) command changes the saved value of RECOVERY TIME LIMIT parameter in Read-Write Error Recovery Mode Page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE(10) command for getting READ RETRY COUNT(N1), WRITE RETRY COUNT(N2) and RECOVERY TIME LIMIT(N3) saved value.<a>
2. Issue MODE SELECT (10) command.
3. Verify expected output <1>
4. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset)
5. Issue MODE SENSE (10) command.<c>
6. Verify expected output<2>.

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h

b) MODE SELECT (10)
PF = 1, SP = 1, Parameter List Length = 14h
AWRE = 1b, READ RETRY COUNT = N1, WRITE RETRY COUNT = N2, RECOVERY TIME LIMIT = N(N != N3) and all other fields set to zero.

c) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 01h, Subpage code = 00h, ALLOCATION LENGTH = 14h</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: AWRE = 1b, READ RETRY COUNT = N1, WRITE RETRY COUNT = N2, RECOVERY TIME LIMIT = N and all other fields are zero.</td></tr>
</table>

7.4.13 Test Case Id: UFS_ModeSelect10_13

This test case may be applied only if the Caching mode page is savable.

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of the MODE SELECT (10) command changes the saved value of WCE parameter in Caching Mode Page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE(10) command for getting WCE(N1) and RCD(N2) saved value.<a>
2. Issue MODE SELECT (10) command.
3. Verify expected output <1>
4. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset)
5. Issue MODE SENSE (10) command.<c>
6. Verify expected output<2>.

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 08h, Subpage code = 00h, ALLOCATION LENGTH = 1Ch

b) MODE SELECT (10)
PF = 1, SP = 1, Parameter List Length = 1Ch
WCE = N(N != N1), RCD = N2 and all other fields set to zero.

c) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 08h, Subpage code = 00h, ALLOCATION LENGTH = 1Ch</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: WCE = N, RCD = N2 and all other fields are zero.</td></tr>
</table>

7.4.14 Test Case Id: UFS_ModeSelect10_14

This test case may be applied only if the Caching mode page is savable.

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of the MODE SELECT (10) command changes the saved value of RCD parameter in Caching Mode Page</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SENSE(10) command for getting WCE(N1) and RCD(N2) saved value.<a>
2. Issue MODE SELECT (10) command.
3. Verify expected output<1>.
4. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset)
5. Issue MODE SENSE (10) command. <c>
6. Verify expected output<2>.

[Clean up]:
1. Restore the original mode page value</td></tr>
<tr><td>Input parameter values</td><td>a) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 08h, Subpage code = 00h, ALLOCATION LENGTH = 1Ch

b) MODE SELECT (10)
PF = 1, SP = 1, Parameter List Length = 1Ch
WCE = N1, RCD = N(N != N2) and all other fields set to zero.

c) MODE SENSE (10)
PC = 11b, LLBAA = 0, DBD = 1,PAGE CODE = 08h, Subpage code = 00h, ALLOCATION LENGTH = 1Ch</td></tr>
<tr><td>Expected Output</td><td>1. MODE SELECT (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. MODE SENSE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
The value of the following fields shall be verified: WCE = N1, RCD = N and all other fields are zero.</td></tr>
</table>

7.4.15 Test Case Id: UFS_ModeSelect10_15

This test case may apply only if the device does not support the PF = 0

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the MODE SELECT (10) command with PF = 0 is terminated with CHECK CONDITION status.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]
None.

[Main]
1. Issue MODE SELECT (10) command.
2. Verify expected output.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SELECT (10)
PF = 0, SP = 0, Parameter List Length = 14h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN CDB</td></tr>
</table></td></tr>
</table>

7.4.16 Test Case Id: UFS_ModeSelect10_16

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the MODE SELECT (10) command with parameter list length = 0</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue MODE SELECT (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SELECT (10)
PF = 1, SP = 0, Parameter List Length = 0</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.4.17 Test Case Id: UFS_ModeSelect10_17

<table>
<tr><td>Ref. specs Section</td><td>UFS: 11.3.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of MODE SELECT command with not supported Page Code value.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue MODE SELECT (10) command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>MODE SELECT (10)
PF = 1, SP = 0, Parameter List Length = 14h
PAGE CODE = InvalidPageCode(1) and all other fields set to zero in the parameter list.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN PARAMETER LIST or NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

NOTE 1 InvalidPageCode value shall be equal to the minimum Page Code value (except zero) not supported by the device. For example, if a device supports only the values:

01h (READ-WRITE ERROR RECOVERY mode page),

08h (CACHING mode page),

0Ah (CONTROL mode page),

the InvalidPageCode is 02h.

The list of supported mode pages may be obtained issuing MODE SENSE (10) command with Page Code set to 3F (ALL PAGES).

7.5 UNMAP Command

The following test cases may be applied to all configured logical unit with bProvisioningType set to 02h or 03h

7.5.1 Test Case Id: UFS_Unmap_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.24</td></tr>
<tr><td>Test Purpose</td><td>To verify device returns CHECK CONDITION status when it receives UNMAP command with UNMAP LOGICAL BLOCK ADDRESS + NUMBER OF LOGICAL BLOCKS which exceeds the logical unit capacity.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. This test case may be executed only if bProvisioningType parameter in the Unit Descriptor is set to 02h or 03h.

[Main]:
1. Issue WRITE (10) command
2. Issue UNMAP command.
3. Verify expected output.<1>
4. Issue READ (10) command
5. Verify expected output.<2>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>WRITE (10)
LBA = RETURNED LOGICAL BLOCK ADDRESS in READ CAPACITY, TRANSFER LENGTH = 01h, WRPROTECT = 000b, DPO = 0b. FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL =00h.
Any Data can be written.

UNMAP CDB
ANCHOR = 0, GROUP NUMBER = 00h, PARAMETER LIST LENGTH = 18h, CONTROL = 00h

UNMAP parameter list
UNMAP DATA LENGTH = 16h
UNMAP BLOCK DESCRIPTOR DATA LENGTH = 10h
UNMAP LOGICAL BLOCK ADDRESS = RETURNED LOGICAL BLOCK ADDRESS in READ CAPACITY,
NUMBER OF LOGICAL BLOCKS = 02h.

READ (10)
LBA = RETURNED LOGICAL BLOCK ADDRESS in READ CAPACITY, TRANSFER LENGTH = 01h, RDPROTECT = 000b, DPO = 0b. FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h,, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE</td></tr>
</table>
2. Compare and verify data buffers of READ (10) and WRITE (10) commands, Data should be same.</td></tr>
</table>

7.5.2 Test Case Id: UFS_Unmap_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.24</td></tr>
<tr><td>Test Purpose</td><td>To verify device behavior when it receives UNMAP command with zero UNMAP BLOCK DESCRIPTOR DATA LENGTH.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]
1. This test case may be executed only if bProvisioningType parameter in the Unit Descriptor is set to 02h or 03h.

[Main]
1. Issue UNMAP command.
2. Verify expected output.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>UNMAP CDB
ANCHOR = 0, GROUP NUMBER = 00h, Parameter List Length = 08h, CONTROL = 00h

UNMAP parameter list
UNMAP DATA LENGTH = 06h
UNMAP BLOCK DESCRIPTOR DATA LENGTH = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.5.3 Test Case Id: UFS_Unmap_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.24</td></tr>
<tr><td>Test Purpose</td><td>To verify device behavior for UNMAP command with NUMBER OF LOGICAL BLOCKS set to zero in UNMAP block descriptor.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]
1. This test case may be executed only if bProvisioningType parameter in the Unit Descriptor is set to 02h or 03h.
[Main]
1. Issue WRITE (10) command.
2. Issue UNMAP command.
3. Verify expected output<1>.
4. Issue READ (10) command.
5. Verify expected output <2>.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>WRITE (10)
LBA = 00h, TRANSFER LENGTH = 01h, WRPROTECT = 000b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.
Any Data can be written.

UNMAP CDB
GROUP NUMBER = 00h, ANCHOR = 0, PARAMETER LIST LENGTH = 18h, CONTROL = 00h

UNMAP
UNMAP DATA LENGTH = 16h,
UNMAP LOGICAL BLOCK ADDRESS = 10h,
UNMAP LOGICAL BLOCK ADDRESS = 00h,
NUMBER OF LOGICAL BLOCKS = 00h

READ (10)
LBA = 00h, TRANSFER LENGTH = 01h, RDPROTECT = 000b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. LBA 0 shall contain the same data before and after the execution of the UNMAP command.</td></tr>
</table>

7.5.4 Test Case Id: UFS_Unmap_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.24</td></tr>
<tr><td>Test Purpose</td><td>To verify that the UNMAP command with PARAMETER LIST LENGTH = 0 terminates with GOOD status.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]
1. This test case may be executed only if bProvisioningType parameter in the Unit Descriptor is set to 02h or 03h.

[Main]
1. Issue UNMAP command.
2. Verify expected output.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>UNMAP CDB
ANCHOR = 0, GROUP NUMBER = 00h, PARAMETER LIST LENGTH = 00h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
</td></tr>
</table>

7.5.5 Test Case Id: UFS_Unmap_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.24</td></tr>
<tr><td>Test Purpose</td><td>To verify that the UNMAP command with ANCHOR = 1 terminates with CHECK CONDITION status.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]
1. This test case may be executed only if bProvisioningType parameter in the Unit Descriptor is set to 02h or 03h.

[Main]
1. Issue WRITE (10) command.
2. Issue UNMAP command.
3. Verify expected output.<1>
4. Issue READ (10) command.
5. Verify expected output.<2>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>WRITE (10)
LBA = 00h, TRANSFER LENGTH = 02h, WRPROTECT = 000b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.
Any Data can be written.

UNMAP CDB
ANCHOR = 1, GROUP NUMBER = 00h, PARAMETER LIST LENGTH = 18h, CONTROL = 00h

UNMAP parameter list
UNMAP DATA LENGTH = 16h,
UNMAP BLOCK DESCRIPTOR DATA LENGTH = 10h,
UNMAP LOGICAL BLOCK ADDRESS = 00h,
NUMBER OF LOGICAL BLOCKS = 02h.

READ (10)
LBA = 00h, TRANSFER LENGTH = 02h, RDPROTECT = 000b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>2. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN CDB</td></tr>
</table>
3. Compare and verify data buffers of READ (10) and WRITE (10) commands, Data should be same.</td></tr>
</table>

7.5.6 Test Case Id: UFS_Unmap_06

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.24</td></tr>
<tr><td>Test Purpose</td><td>To verify that device returns GOOD status when a block descriptor with odd size (not multiple of 16) is passed for UNMAP command</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. This test case may be executed only if bProvisioningType parameter in the Unit Descriptor is set to 02h or 03h.

[Main]:
1. Issue WRITE (10) command.
2. Issue UNMAP command.
3. Verify expected output<1>.
4. Issue READ (10) command.
5. Verify expected output<2>.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>WRITE (10)
LBA = 00h, TRANSFER LENGTH = 02h, WRPROTECT = 0b, DPO = 0b,
FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.
Any Data can be written.

UNMAP CDB
ANCHOR = 0, GROUP NUMBER = 00h, PARAMETER LIST LENGTH = 14h,
CONTROL = 00h

UNMAP parameter list
UNMAP DATA LENGTH = 12h
UNMAP BLOCK DESCRIPTOR DATA LENGTH = 0Ch,
UNMAP LOGICAL BLOCK ADDRESS = 00h,
NUMBER OF LOGICAL BLOCKS = 02h

READ (10)
LBA = 00h, TRANSFER LENGTH = 02h, WRPROTECT = 0b, DPO = 0b,
FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. LBA 0 shall contain the same data before and after execution of the UNMAP command.</td></tr>
</table>

7.6 START STOP UNIT Command

7.6.1 Test Case Id: UFS_StartStopUnit_01

This test may be applied to all configured logical unit (LUN field in UPIU: from 00h to 07h).

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.10</td></tr>
<tr><td>Test Purpose</td><td>To verify START STOP UNIT command with START = 0, IMMED bit = 0.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]
1. Issue START STOP UNIT command.
2. Verify expected output.<1>
3. Issue TEST UNIT READY command
4. Verify expected output.<2>

[Clean up]:
1. Issue a START STOP UNIT command with START = 1</td></tr>
<tr><td>Input parameter values</td><td>START STOP UNIT
IMMED = 0, POWER CONDITION = 0h, POWER CONDITION MODIFIER = 0h, START = 0, NO_FLUSH = 0, LOEJ = 0, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The START STOP UNIT command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The TEST UNIT READY command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>NOT READY</td><td>-</td></tr>
</table></td></tr>
</table>

7.6.2 Test Case Id: UFS_StartStopUnit_02

This test may be applied to all configured logical unit (LUN field in UPIU: from 00h to 07h).

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.10</td></tr>
<tr><td>Test Purpose</td><td>To verify START STOP UNIT command with START bit = 1, IMMED bit = 0.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue a START STOP UNIT command with START = 0 (see –a-).
2. Verify that the command executed with GOOD status.
3. Issue TEST UNIT READY command
4. Check terminates with CHECK CONDITION status, with the sense key set to NOT READY.
[Main]:
1. Issue START STOP UNIT command. (see –b-).
2. Verify expected output.<1>
3. Issue TEST UNIT READY command
4. Verify expected output.<2>
[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) START STOP UNIT
IMMED = 0,POWER CONDITION = 0h, POWER CONDITION MODIFIER = 0h, NO_FLUSH = 0, LOEJ = 0, START = 0, CONTROL = 00h
b) START STOP UNIT
IMMED = 0,POWER CONDITION = 0h, POWER CONDITION MODIFIER = 0h, NO_FLUSH = 0, LOEJ = 0, START = 1, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The START STOP UNIT command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The TEST UNIT READY command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.6.3 Test Case Id: UFS_StartStopUnit_03

This test may be applied to all configured logical unit.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.10</td></tr>
<tr><td>Test Purpose</td><td>To verify that START STOP UNIT command with START = 0, IMMED bit = 1.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue START STOP UNIT command.
2. Verify expected output.

[Clean up]:
1. Issue a START STOP UNIT command with START =1, IMMED bit = 0</td></tr>
<tr><td>Input parameter values</td><td>START STOP UNIT
IMMED = 1,POWER CONDITION = 0h, POWER CONDITION MODIFIER = 0, START = 0, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The START STOP UNIT command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
</td></tr>
</table>

7.6.4 Test Case Id: UFS_StartStopUnit_04

This test may be applied to all configured logical unit (LUN field in UPIU: from 00h to 07h).

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.10</td></tr>
<tr><td>Test Purpose</td><td>To verify START STOP UNIT command with START bit = 1, IMMED bit = 1.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue a START STOP UNIT command with START = 0, IMMED bit = 0 (see –a-).
2. Verify that the command executed with GOOD status.
3. Issue TEST UNIT READY command
4. Check terminates with CHECK CONDITION status, with the sense key set to NOT READY.
[Main]:
1. Issue START STOP UNIT command. (see –b-)
2. Verify expected output.
[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) START STOP UNIT
IMMED = 0,POWER CONDITION = 0h, POWER CONDITION MODIFIER = 0h, NO_FLUSH = 0, LOEJ = 0, START = 0, CONTROL = 00h

b) START STOP UNIT
IMMED = 1,POWER CONDITION = 0h, POWER CONDITION MODIFIER = 0h, NO_FLUSH = 0, LOEJ = 0, START = 1, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The START STOP UNIT command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.6.5 Test Case Id: UFS_StartStopUnit_05

This test may be applied to the UFS Device well-know logical unit only (LUN field in UPIU = D0h)..

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.10</td></tr>
<tr><td>Test Purpose</td><td>To verify START STOP UNIT command with POWER CONDITION = 2h (UFS Sleep Mode).</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]
1. Issue START STOP UNIT command.
2. Verify expected output.<1>
3. Issue QUERY REQUEST UPIU to read bCurrentPowerMode Attribute
4. Verify expected output<2>.

[Clean up]:
1. Issue START STOP UNIT command and change the device state to active. (IMMED = 0,POWER CONDITION = 1h, POWER CONDITION MODIFIER = 0h, START = 0)</td></tr>
<tr><td>Input parameter values</td><td>START STOP UNIT
IMMED = 0,POWER CONDITION = 2h, POWER CONDITION MODIFIER = 0h, START = 0, Control = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The START STOP UNIT command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. bCurrentPowerMode Attribute shall be set to UFS-Sleep mode</td></tr>
</table>

7.6.6 Test Case Id: UFS_StartStopUnit_06

This test may be applied to the UFS Device well-know logical unit only (LUN field in UPIU = D0h).

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.10</td></tr>
<tr><td>Test Purpose</td><td>To verify START STOP UNIT command with POWER CONDITION = 3h (UFS Power Down Mode).</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue a START STOP UNIT command.
2. Verify expected output.<1>
3. Issue a QUERY REQUEST UPIU to read bCurrentPowerMode Attribute
4. Verify expected output.<2>

[Clean up]:
1. Issue START STOP UNIT command to change the power mode to Active. (IMMED = 0,POWER CONDITION = 1h, POWER CONDITION MODIFIER = 0h, START = 0)</td></tr>
<tr><td>Input parameter values</td><td>START STOP UNIT
IMMED = 0,POWER CONDITION = 3h, POWER CONDITION MODIFIER = 0h, START = 0, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The START STOP UNIT command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. bCurrentPowerMode Attribute shall be set to UFS-PowerDown mode</td></tr>
</table>

7.6.7 Test Case Id: UFS_StartStopUnit_07

This test may be applied to UFS Device well-know logical unit only (LUN field in UPIU = D0h).

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.10</td></tr>
<tr><td>Test Purpose</td><td>To verify START STOP UNIT command with invalid POWER CONDITION value.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue a START STOP UNIT command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>START STOP UNIT
IMMED = 0, POWER CONDITION = 0Ch, POWER CONDITION MODIFIER = 0h, START = 0, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN CDB or NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.6.8 Test Case Id: UFS_StartStopUnit_08

This test may be applied to Boot well-know logical unit only (LUN field in UPIU = B0h).

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.10 and 10.6.6</td></tr>
<tr><td>Test Purpose</td><td>To verify that the START STOP UNIT command is an invalid command for the Boot well-known logical unit</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]
1. Issue START STOP UNIT command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>START STOP UNIT
IMMED = 0,POWER CONDITION = 0h, POWER CONDITION MODIFIER = 0, START = 1, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID COMMAND OPERATION CODE.</td></tr>
</table></td></tr>
</table>

7.7 READ CAPACITY (10) Command

The following test cases may be applied to all configured logical units.

7.7.1 Test Case Id: UFS_ReadCapacity10_01

Obsolete

7.7.2 Test Case Id: UFS_ReadCapacity10_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.8</td></tr>
<tr><td>Test Purpose</td><td>To verify that the READ CAPACITY (10) command is terminated with GOOD status. .</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue a READ CAPACITY (10) command.
2. Verify expected output.<1>
3. Issue a QUERY REQUEST UPIU to read bLogicalBlockSize and qLogicalBlockCount in UNIT descriptor.
4. Verify expected output.<2>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>READ CAPACITY (10)
LOGICAL BLOCK ADDRESS = 0, PMI = 0b, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. RETURNED LOGICAL BLOCK ADDRESS shall be equal to qLogicalBlockCount – 1 and LOGICAL BLOCK LENGTH in Byte shall be equal to $2^{bLogicalBlockSize}$.</td></tr>
</table>

7.7.3 Test Case Id: UFS_ReadCapacity10_03

Obsolete

7.8 FORMAT UNIT Command

The following test cases may be applied to all configured logical units.

7.8.1 Test Case Id: UFS_FormatUnit_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.18</td></tr>
<tr><td>Test Purpose</td><td>To verify status of FORMAT UNIT command with FMTPINFO and FMTDATA set to zero.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue a FORMAT UNIT command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>FORMAT UNIT
FMTPINFO = 00b, LONGLIST = 0, FMTDATA = 0, CMPLIST = 0,
DEFECT LIST FORMAT = 0h, CONTROL=00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.8.2 Test Case Id: UFS_FormatUnit_02

7.8.3 Test Case Id: UFS_FormatUnit_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.18 and 12.2.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the response for Format Unit command when Power On write protection is enabled and fPowerOnWPEn is 1.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. bLUWriteProtect parameter in Unit descriptor shall be set to 01h
2. Write some data using WRITE (10) command to LBA 0, check the return status, it should be GOOD.

[Main]:
1. Enable the Power On write protection by setting the fPowerOnWPEn to 1
2. Issue a FORMAT UNIT command.
3. Verify expected output.<1>
4. Issue a READ (10) command to the same LBA used in WRITE (10) before formatting
5. Verify expected output.<2>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>FORMAT UNIT
FMTPINFO = 0 , FMTDATA = 0, LONGLIST = 0, CMPLST = 0,
DEFECT LIST FORMAT = 0h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>DATA PROTECT</td><td>LUN WRITE PROTECT or NO ADDITIONAL SENSE INFORMATION</td></tr>
</table>
2. Read data shall match with the written data before FORMAT UNIT command is issued.</td></tr>
</table>

7.8.4 Test Case Id: UFS_FormatUnit_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.18 and 12.2.3.4</td></tr>
<tr><td>Test Purpose</td><td>To verify the response for Format Unit command when Permanent write protection is enabled and fPermanentWPEn is 1.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. bLUWriteProtect parameter in Unit descriptor shall be set to 02h
2. Write some data using WRITE (10) command to LBA 0, check the return status, it should be GOOD.

[Main]:
1. Enable the Power On write protection by setting the fPermanentWPEn to 1
2. Issue a FORMAT UNIT command.
3. Verify expected output.<1>
4. Issue a READ (10) command to the same LBA used in WRITE (10) before formatting
5. Verify expected output.<2>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>FORMAT UNIT
FMTPINFO = 0, FMTDATA = 0, LONGLIST = 0, CMPLST = 0,
DEFECT LIST FORMAT = 0h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table><tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr><tr><td>Target Failure</td><td>CHECK CONDITION</td><td>DATA PROTECT</td><td>LUN WRITE PROTECT
or
NO ADDITIONAL SENSE INFORMATION</td></tr></table>2. Data read should match with the data written before FORMAT UNIT command is issued.</td></tr>
</table>

7.9 TEST UNIT READY Command

7.9.1 Test Case Id: UFS_TestUnitReady_01

The following test cases may be applied to all configured logical units and well-known logical units.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.11</td></tr>
<tr><td>Test Purpose</td><td>To verify the TEST UNIT READY command when sent after the logical unit is accepting medium access command.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue Read 10 command.
2. Verify that the command terminates with GOOD STATUS.

[Main]:
1. Issue TEST UNIT READY command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>TEST UNIT READY
CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.9.2 Test Case Id: UFS_TestUnitReady_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.11</td></tr>
<tr><td>Test Purpose</td><td>To verify the TEST UNIT READY command with invalid LUN is terminated with CHECK CONDITION status.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue TEST UNIT READY command.
2. Verify expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>TEST UNIT READY
CONTROL = 00h, LUN in UPIU = 09h</td></tr>
<tr><td>Expected Output</td><td>1. The command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL UNIT NOT SUPPORTED or NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.10 WRITE (6) Command

The following test cases may be applied to all configured logical units.

7.10.1 Test Case Id: UFS_Write6_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.14</td></tr>
<tr><td>Test Purpose</td><td>To verify WRITE (6) command writes 256 block of data when TRANSFER LENGTH = 00h.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Send READ CAPACITY (10) command and verify LU has more than or equal to 256 Logical blocks

[Main]:
1. Issue READ (6) command.<a>
2. Issue WRITE (6) command.
3. Verify Expected Output.<1>
4. Issue a READ (6) command.
5. Verify Expected Output.<2>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) READ (6)
LBA = 00h, TRANSFER LENGTH = 00h, CONTROL = 00h

WRITE (6)
LBA = 00h, TRANSFER LENGTH = 00h, CONTROL = 00h.
Any Data can be written except that is read from READ (6) <a> command.

b) READ (6)
LBA = 00h, TRANSFER LENGTH = 00h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The WRITE (6) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Compare data buffers of READ (6) and WRITE (6) commands. Data shall be same.</td></tr>
</table>

7.10.2 Test Case Id: UFS_Write6_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.14</td></tr>
<tr><td>Test Purpose</td><td>To verify WRITE (6) command when LBA plus TRANSFER LENGTH exceeds the capacity of the medium</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue READ CAPACITY (10) command to find the Max addressable LBA(RETURN LOGICAL BLOCK ADDRESS in Parameter Data) of the LU and it should be less than (2^{16}) Logical blocks.

[Main]
1. Issue READ (6) command.<a>
2. Issue WRITE (6) command.
3. Verify the Expected Output.<1>
4. Issue READ (6) command.
5. Verify the Expected Output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) READ (6)
LBA = Max addressable LBA, TRANSFER LENGTH = 01h, CONTROL = 00h.

WRITE (6)
LBA = Max addressable LBA, TRANSFER LENGTH = 02h, CONTROL = 00h.
Any Data can be written.

b) READ (6)
LBA = Max addressable LBA, TRANSFER LENGTH = 01h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The WRITE (6) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table>
2. Compare data buffer of READ (6) command and READ (6)<a> command, Data shall be same.</td></tr>
</table>

7.10.3 Test Case Id: UFS_Write6_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.14</td></tr>
<tr><td>Test Purpose</td><td>To verify WRITE (6) command when LBA exceeds the capacity of the medium</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue READ CAPACITY (10) command to find the Max addressable LBA(RETURN LOGICAL BLOCK ADDRESS in Parameter Data) of the LU, and it should be less than (2^{16} - 1) Logical blocks
[Main]:
1. Issue WRITE (6) command.
2. Verify the Expected Output.
[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>WRITE (6)
LBA = Max addressable LBA + 1, TRANSFER LENGTH = 00h, CONTROL = 00h.
Any Data can be written.</td></tr>
<tr><td>Expected Output</td><td>1. The WRITE (6) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.11 WRITE (10) Command

The following test cases may be applied to all configured logical units.

7.11.1 Test Case Id: UFS_Write10_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.15</td></tr>
<tr><td>Test Purpose</td><td>To verify the WRITE (10) command with LBA = 00h and valid transfer length</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue READ (10) command.<a>
2. Issue WRITE (10) command.
3. Verify Expected Output.<1>
4. Issue READ (10) command.
5. Verify Expected Output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) READ (10)
LBA = 00h, TRANSFER LENGTH = 04h, RDPROTECT = 0b, DPO = 0b,
FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.

WRITE (10)
LBA = 00h, TRANSFER LENGTH = 04h, WRPROTECT = 0b, DPO = 0b,
FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.
Any Data can be written except that is read from READ (10) <a> command.

b) READ (10)
LBA = 00h, TRANSFER LENGTH = 04h, RDPROTECT = 0b, DPO = 0b,
FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The WRITE (10) command response shall be:
<table><tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr><tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr></table>2. Compare data buffers of READ (10) and WRITE (10) commands, Data shall be same.</td></tr>
</table>

7.11.2 Test Case Id: UFS_Write10_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.15</td></tr>
<tr><td>Test Purpose</td><td>To verify WRITE (10) command when LBA plus TRANSFER LENGTH exceeds the capacity of the medium</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue READ CAPACITY (10) command to find the Max addressable LBA (RETURN LOGICAL BLOCK ADDRESS in Parameter Data) of the LU.
2. Issue a READ (10) command.<a>
3. Issue a WRITE (10) command.
4. Verify the Expected Output.<1>
5. Issue a READ (10) command.
6. Verify the Expected Output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) READ (10)
LBA = Max addressable LBA, TRANSFER LENGTH = 01h, RDPROTECT = 0b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL =00h.

WRITE (10)
LBA = Max addressable LBA, TRANSFER LENGTH = 02h, WRPROTECT = 0b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL =00h.
Any Data can be written except that is read from READ (10) <a> command.

b) READ (10)
LBA = Max addressable LBA, TRANSFER LENGTH = 01h, RDPROTECT = 0b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL =00h.</td></tr>
<tr><td>Expected Output</td><td>1. The WRITE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table>
2. Compare data buffer of READ (10) and READ (10)<a> commands, Data shall be same.</td></tr>
</table>

7.11.3 Test Case Id: UFS_Write10_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.15</td></tr>
<tr><td>Test Purpose</td><td>To verify WRITE (10) command when LBA exceeds the capacity of the medium.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue READ CAPACITY (10) command to find the Max addressable LBA (RETURN LOGICAL BLOCK ADDRESS in Parameter Data) of the LU.
2. Issue a WRITE (10) command.
3. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>WRITE (10)
LBA = Max addressable LBA + 1, TRANSFER LENGTH = 00h, WRPROTECT = 0b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The WRITE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.11.4 Test Case Id: UFS_Write10_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.15</td></tr>
<tr><td>Test Purpose</td><td>To verify WRITE (10) command with FUA = 1b, DPO = 0b.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue READ (10) command.<a>
2. Issue WRITE (10) Command.
3. Verify the Expected Output.<1>
4. Issue READ (10) command.
5. Verify the Expected Output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) READ (10)
LBA = 00h, TRANSFER LENGTH = 01h, RDPROTECT = 0b, DPO = 0b,
FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.

WRITE (10)
LBA = 00h, TRANSFER LENGTH = 01h, WRPROTECT = 0b, DPO = 0b,
FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.
Any Data can be written except that is read from READ (10) <a> command.

b) READ (10)
LBA = 00h, TRANSFER LENGTH = 01h, RDPROTECT = 0b, DPO = 0b,
FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The WRITE (10) command response shall be:
<table><tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr><tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr></table>2. Compare data buffers of READ (10) and WRITE (10) commands, Data shall be same.</td></tr>
</table>

7.11.5 Test Case Id: UFS_Write10_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.15</td></tr>
<tr><td>Test Purpose</td><td>To verify WRITE (10) command with DPO = 1b, FUA = 1b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue READ (10) command.<a>
2. Issue WRITE (10) Command.
3. Verify the Expected Output.<1>
4. Issue READ (10) command.
5. Verify the Expected Output.<2>

[Cleanup]:
None.</td></tr>
<tr><td>Input parameter values</td><td>a) READ (10)
LBA = 00h, TRANSFER LENGTH = 01h, RDPROTECT = 0b, DPO = 1b, FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.

WRITE (10)
LBA = 00h, TRANSFER LENGTH = 01h, WRPROTECT = 0b, DPO = 1b. FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.
Any Data can be written except that is read from READ (10) <a> command.

b) READ (10)
LBA = 00h, TRANSFER LENGTH = 01h, RDPROTECT = 0b, DPO = 1b. FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The WRITE (10) command response shall be:
<table><tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr><tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr></table>2. Compare data buffers of READ (10) and WRITE (10) commands, Data shall be same.</td></tr>
</table>

7.11.6 Test Case Id: UFS_Write10_06

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.15</td></tr>
<tr><td>Test Purpose</td><td>To verify the WRITE (10) command with valid LBA and TRANSFER LENGTH = 00h</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue READ (10) command.<a>
2. Issue WRITE (10) command.
3. Verify the Expected Output.<1>
4. Issue READ (10) command.
5. Verify the Expected Output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) READ (10)
LBA =00h, TRANSFER LENGTH =01h, RDPROTECT = 0b, DPO = 0b, FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.

WRITE (10)
LBA =00h, TRANSFER LENGTH =00h, WRPROTECT = 0b, DPO = 0b, FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.

b) READ (10)
LBA =00h, TRANSFER LENGTH =01h, RDPROTECT = 0b, DPO = 0b, FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The WRITE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Compare data buffers of READ (10) and READ (10)<a> commands, Data shall be same.</td></tr>
</table>

7.12 Read (6) Command

The following test cases may be applied to all configured logical units and BOOT well-known logical unit if it is enabled.

7.12.1 Test Case Id: UFS_Read6_01

If this test case is used to BOOT well-known logical unit, WRITE command may not be issued from host because WRITE command is not supported on BOOT well-known logical unit. Data comparison may not be processed.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.5</td></tr>
<tr><td>Test Purpose</td><td>To Verify the READ (6) command reads 256 blocks of data when TRANSFER LENGTH = 00h.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue READ CAPACITY (10) command and verify LU has more than or equal to 256 Logical blocks

[Main]:
1. Issue WRITE (6) command.
2. Issue READ (6) command.
3. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>WRITE (6)
LBA = 00h, TRANSFER LENGTH = 00h, CONTROL = 00h.
Any Data can be written.

READ (6)
LBA = 00h, TRANSFER LENGTH = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The READ (6) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Compare data buffers of WRITE (6) and READ (6) commands. Data shall be same.</td></tr>
</table>

7.12.2 Test Case Id: UFS_Read6_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.5</td></tr>
<tr><td>Test Purpose</td><td>To verify READ (6) command when LBA plus TRANSFER LENGTH exceeds the capacity of the medium</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue READ CAPACITY (10) command to find Logical Block count (Max addressable LBA) of the LU and it should not be greater than or equal to (2^{16}) Logical Blocks
[Main]:
1. Issue READ (6) command.
2. Verify the Expected Output.
[Cleanup] :
None</td></tr>
<tr><td>Input parameter values</td><td>READ (6)
LBA = Maximum addressable LBA, TRANSFER LENGTH = 02h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The READ (6) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST-</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.12.3 Test Case Id: UFS_Read6_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.5</td></tr>
<tr><td>Test Purpose</td><td>To verify READ (6) command when LBA exceeds the capacity of the medium</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue READ CAPACITY (10) command to find the Logical Block count (MAX addressable LBA) of the LU, and it should not be greater than or equal to (2^{16} - 1) logical blocks
[Main]:
1. Issue READ (6) command.
2. Verify the Expected Output.
[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>READ (6)
LBA = MAX addressable LBA + 1, TRANSFER LENGTH = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The READ (6) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST-</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.13 Read (10) Command

The following test cases may be applied to all configured logical units and BOOT well-known logical unit if it is enabled.

If following test cases are used to BOOT well-known logical unit, WRITE command may not be issued from host because WRITE command is not supported on BOOT well-known logical unit. Data comparison may not be processed.

7.13.1 Test Case Id: UFS_Read10_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.6</td></tr>
<tr><td>Test Purpose</td><td>To verify the READ (10) command with valid LBA and TRANSFER LENGTH.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue WRITE (10) command.
2. Issue READ (10) command.
3. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>WRITE (10)
LBA = 00h, TRANSFER LENGTH = 04h, WRPROTECT = 0b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.
Any Data can be written.

READ (10)
LBA = 00h, TRANSFER LENGTH = 04h, RDPROTECT = 0b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The READ (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Compare data buffers of READ (10) and WRITE (10) commands. Data shall be same.</td></tr>
</table>

7.13.2 Test Case Id: UFS_Read10_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.6</td></tr>
<tr><td>Test Purpose</td><td>To verify READ (10) command when LBA plus TRANSFER LENGTH exceeds the capacity of the medium</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue READ CAPACITY (10) command to find out the Max addressable LBA (RETURN LOGICAL BLOCK ADDRESS in Parameter Data).
2. Issue READ (10) command.
3. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>READ (10)
LBA = Max addressable LBA, TRANSFER LENGTH = 02h, RDPROTECT = 0b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The READ (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.13.3 Test Case Id: UFS_Read10_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.6</td></tr>
<tr><td>Test Purpose</td><td>To verify READ (10) command when LBA exceeds the capacity of the medium</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None
[Main]:
1. Issue READ CAPACITY (10) command to find out the Max addressable LBA (RETURN LOGICAL BLOCK ADDRESS in Parameter Data).
2. Issue READ (10) command.
3. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>READ (10)
LBA = Max addressable LBA + 1, TRANSFER LENGTH = 00h, RDPROTECT = 0b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL =00h.</td></tr>
<tr><td>Expected Output</td><td>1. The READ (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.13.4 Test Case Id: UFS_Read10_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.6</td></tr>
<tr><td>Test Purpose</td><td>To verify READ (10) command with FUA = 1, DPO = 0b.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue WRITE (10) command.
2. Issue READ (10) command.
3. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>WRITE (10)
LBA = 00h, TRANSFER LENGTH = 01h, WRPROTECT = 0b, DPO = 0b, FUA = 1, FUA_NV = 0, GROUP NUMBER = 00h, CONTROL = 00h.
Any Data can be written.

READ (10)
LBA = 00h, TRANSFER LENGTH = 01h, RDPROTECT = 0b, DPO = 0b, FUA = 1, FUA_NV = 0, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The READ (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Compare data buffers of READ (10) and WRITE (10) commands, Data shall be same.</td></tr>
</table>

7.13.5 Test Case Id: UFS_Read10_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.6</td></tr>
<tr><td>Test Purpose</td><td>To verify READ (10) command with DPO = 1b, FUA = 1b.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue WRITE (10) command.
2. Issue READ (10) command.
3. Verify the Expected Output.
.
[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>WRITE (10)
LBA = 00h, TRANSFER LENGTH = 01h, WRPROTECT = 0b, DPO = 1b, FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.

READ (10)
LBA = 00h, TRANSFER LENGTH = 01h, RDPROTECT = 0b, DPO = 1b, FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The READ (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Compare data buffers of READ (10) and WRITE (10) commands, Data shall be same.</td></tr>
</table>

7.13.6 Test Case Id: UFS_Read10_06

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.6</td></tr>
<tr><td>Test Purpose</td><td>To verify the READ (10) command with valid LBA and TRANSFER LENGTH = 00h.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]
1. Issue READ (10) command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>READ (10)
LBA = 00h, TRANSFER LENGTH = 00h, RDPROTECT = 0b, DPO = 0b, FUA = 1b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL =00h.</td></tr>
<tr><td>Expected Output</td><td>1. The READ (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
If the requested data size is 0, then there is no data phase</td></tr>
</table>

7.14 Verify (10) Command

The following test cases may be applied to all configured logical units.

7.14.1 Test Case Id: UFS_Verify10_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.13</td></tr>
<tr><td>Test Purpose</td><td>To verify the VERIFY (10) command with BYTCHK = 0b.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue WRITE (10) command to LBA from 0 to 0Ah.

[Main]:
1. Issue VERIFY (10) command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>VERIFY (10)
LBA = 00h, VERIFICATION LENGTH = 0Ah, VRPROTECT = 000b, DPO = 0b, BYTCHK = 0b, GROUP NUMBER = 0h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The VERIFY (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
</td></tr>
</table>

7.14.2 Test Case Id: UFS_Verify10_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.13</td></tr>
<tr><td>Test Purpose</td><td>To verify the VERIFY (10) command with VERIFICATION LENGTH = 00h.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue VERIFY (10) command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>VERIFY (10)
LBA = 00h, VERIFICATION LENGTH = 00h, VRPROTECT = 000b, DPO = 0b, BYTCHK = 0b, GROUP NUMBER = 0h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The VERIFY (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
</td></tr>
</table>

7.14.3 Test Case Id: UFS_Verify10_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.13</td></tr>
<tr><td>Test Purpose</td><td>To verify the VERIFY (10) command with LBA plus the VERIFICATION LENGTH exceeds the capacity of the medium.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]
1. Issue READ CAPACITY (10) command to find the Max addressable LBA (RETURNED LOGICAL BLOCK ADDRESS in Parameter Data).

[Main]
1. Issue VERIFY (10) command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter value</td><td>VERIFY (10)
LBA = Max addressable LBA, VERIFICATION LENGTH = 02h, VRPROTECT = 000b, DPO = 0b, BYTCHK = 0b, GROUP NUMBER = 0h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The VERIFY (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.14.4 Test Case Id: UFS_Verify10_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.13</td></tr>
<tr><td>Test Purpose</td><td>To verify the VERIFY (10) command with LBA exceeds the capacity of the medium.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue READ CAPACITY (10) command to find the Max addressable LBA (RETURNED LOGICAL BLOCK ADDRESS in Parameter Data).
[Main]:
1. Issue VERIFY (10) command.
2. Verify the Expected Output.
[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>VERIFY (10)
LBA = Max addressable LBA + 1, VERIFICATION LENGTH = 00h, VRPROTECT = 000b, DPO = 0b, BYTCHK = 0b, GROUP NUMBER = 0h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The VERIFY (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.15 Send Diagnostic Command

The following test cases may be applied to all configured logical units

7.15.1 Test Case Id: UFS_SendDiagnostic_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.21</td></tr>
<tr><td>Test Purpose</td><td>To verify default self test feature of SEND DIAGNOSTIC command is support by device</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue the SEND DIAGNOSTIC command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>SEND DIAGNOSTIC
SELFTEST = 1b, SELF-TEST CODE = 0, PF = 0b, DEVOFFL = 0b, UNITOFFL = 0b, Parameter List Length = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The SEND DIAGNOSTIC command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.15.2 Test Case Id: UFS_SendDiagnostic_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.21</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of SEND DIAGNOSTIC command with SELFTEST = 0b, PF = 0b, SELF-TEST CODE = 000b and Parameter List Length = 00h</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue the SEND DIAGNOSTIC command..
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>SEND DIAGNOSTIC
SELFTEST = 0b, SELF-TEST CODE = 0, PF =0b, DEVOFFL = 0b, UNITOFFL = 0b, Parameter List Length = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The SEND DIAGNOSTIC command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.15.3 Test Case Id: UFS_SendDiagnostic_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.21</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of SEND DIAGNOSTIC command with SELF TEST = 1b and Parameter List Length is not zero</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue the SEND DIAGNOSTIC command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>SEND DIAGNOSTIC
SELFTEST = 1b, SELF-TEST CODE = 0, PF = 0b, DEVOFFL = 0b, UNITOFFL = 0b, Parameter List Length = FFh, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The SEND DIAGNOSTIC command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN CDB or NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.16 REPORT LUNS Command

7.16.1 Test Case Id: UFS_ReportLuns_01

The following test cases may be applied to all configured logical units and REPORT LUNS well-known logical unit only.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.12</td></tr>
<tr><td>Test Purpose</td><td>To verify that REPORT LUNS command is working on Device by getting the Report of the regular Logical units.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue Query request Read descriptor command and read the number of LUNs from the Device descriptor.

[Main]:
1. Issue the REPORT LUNS command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter value</td><td>REPORT LUNS
SELECT REPORT = 00h, ALLOCATION LENGTH = (number of LUNs *8)+ 8, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The REPORT LUNS command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Format should be in Peripheral Device Addressing Format</td></tr>
</table>

7.16.2 Test Case Id: UFS_ReportLuns_02

The following test cases may be applied to all configured logical units and REPORT LUNS well-known logical unit only.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.12</td></tr>
<tr><td>Test Purpose</td><td>To verify that REPORT LUNS command is working on Device by getting the Report of the well known logical units.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue the REPORT LUNS command.
2. Verify the Expected Output

[Cleanup]:
None</td></tr>
<tr><td>Input parameter value</td><td>REPORT LUNS
SELECT REPORT = 01h, ALLOCATION LENGTH = 28h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The REPORT LUNS command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. LUN LIST LENGTH shall be 32 and the LUN addresses shall be among the 01h, 50h, 30h and 44h.

3. Format should be in Well Known Logical Unit Extended Addressing Format</td></tr>
</table>

7.16.3 Test Case Id: UFS_ReportLuns_03

The following test cases may be applied to all configured logical units and REPORT LUNS well-known logical unit only.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.12</td></tr>
<tr><td>Test Purpose</td><td>To verify that REPORT LUNS command is working on Device by getting the Report, containing details of regular and Well known logical units together.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue Query request Read descriptor command and read the number of LUNs from the Device descriptor.

[Main]:
1. Issue the REPORT LUNS command.
2. Verify the Expected Output.

[Cleanup]:
None

Note:
NumWLUN i.e. Number of Well-known LUNs = 4 as per the UFS Spec.</td></tr>
<tr><td>Input parameter value</td><td>REPORT LUNS
SELECT REPORT = 02h, ALLOCATION LENGTH = ((NumLUNs + NumWLUN)*8)+8, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The REPORT LUNS command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Verify the return data of command (LUN LIST LENGTH should be multiplied by eight, i.e. ((No of Regular + No of W-LUN's)*8).

3. Format should be in Peripheral Device Addressing Format for General Purpose LUNs and Well Known Logical Unit Extended Addressing Format for Well Known LUNs</td></tr>
</table>

7.16.4 Test Case Id: UFS_ReportLuns_04

The following test cases may be applied to all configured logical units and REPORT LUNS well-known logical unit only.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.12</td></tr>
<tr><td>Test Purpose</td><td>To verify the REPORT LUNS command when the ALLOCATION LENGTH field is less than 16</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue the REPORT LUNS command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>REPORT LUNS
SELECT REPORT = 01h, ALLOCATION LENGTH = 0Fh, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The REPORT LUNS command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST-</td><td>INVALID FIELD IN CDB
or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.16.5 Test Case Id: UFS_ReportLuns_05

The following test cases may be applied to all configured logical units and REPORT LUNS well-known logical unit only.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.12</td></tr>
<tr><td>Test Purpose</td><td>To verify the REPORT LUNS command with invalid SELECT REPORT.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue the REPORT LUNS command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>REPORT LUNS
SELECT REPORT = 03h(Reserved), ALLOCATION LENGTH = 68h, CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The REPORT LUNS command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST-</td><td>INVALID FIELD IN CDB
or
NO ADDITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.16.6 Test Case Id: UFS_ReportLuns_06

This test may be applied to UFS Device well-known logical unit only (LUN field in UPIU = D0h), RPMB well-known logical unit, Boot well-known logical unit.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.6.6 and 11.3.12</td></tr>
<tr><td>Test Purpose</td><td>To verify the REPORT LUNS command is an invalid command for the UFS Device well-known logical unit.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue the REPORT LUNS command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>REPORT LUNS
SELECT REPORT = 01h, ALLOCATION LENGTH = 20h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The REPORT LUNS command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST-</td><td>INVALID COMMAND OPERATION CODE or NO ADITIONAL SENSE INFORMATION</td></tr>
</table></td></tr>
</table>

7.16.7 Test Case Id: UFS_ReportLuns_07

The following test cases may be applied to all configured logical units and REPORT LUNS well-known logical unit only.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.12</td></tr>
<tr><td>Test Purpose</td><td>To verify REPORT LUNS Command with Pending UNIT ATTENTION condition.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
2. Execute hardware reset to establish UNIT ATTENTION condition and not send REQUEST SENSE command.
[Main]:
1. Issue the REPORT LUNS Command.
2. Verify the Expected Output.
[Clean up]:
1. Send REQUEST SENSE command to clear UNIT ATTENTION condition.</td></tr>
<tr><td>Input parameter values</td><td>REPORT LUNS
SELECT REPORT = 01h, ALLOCATION LENGTH = 28h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The REPORT LUNS command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. LUN LIST LENGTH shall be 32 and the LUN addresses shall be among the 01h, 50h, 30h and 44h.
3. Data response shall be in Well Known Logical Unit Extended Addressing Format</td></tr>
</table>

7.16.8 Test Case Id: UFS_ReportLuns_08

The following test cases may be applied to all configured logical units and REPORT LUNS well-known logical unit only.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.12</td></tr>
<tr><td>Test Purpose</td><td>To verify REPORT LUNS command when the number of data bytes transferred by the Device server is less than ALLOCATION LENGTH.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue Query request Read descriptor command and read the number of LUNs from the Device descriptor.

[Main]:
1. Issue the REPORT LUNS command.
2. Verify the Expected Output.

[Cleanup]:
None.</td></tr>
<tr><td>Input parameter values</td><td>REPORT LUNS
SELECT REPORT = 00h, ALLOCATION LENGTH = (number of LUNs *8) + 16</td></tr>
<tr><td>Expected Output</td><td>1. The REPORT LUNS command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. Verify the return data of command (LUN LIST LENGTH should be multiplied by eight i.e. (number of LUNs *8)).

3. The data bytes transferred from device server = (number of LUNs * 8) +8.

4. Under flow flag in Response UPIU shall be set to 1b.</td></tr>
</table>

7.16.9 Test Case Id: UFS_ReportLuns_09

The following test cases may be applied to all configured logical units and REPORT LUNS well-known logical unit only.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.12</td></tr>
<tr><td>Test Purpose</td><td>To verify REPORT LUNS command when the actual number of data bytes to be transferred by the device server is greater than ALLOCATION LENGTH.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue the REPORT LUNS Command.
2. Verify Expected Output.

[Cleanup]:
None.</td></tr>
<tr><td>Input parameter value</td><td>REPORT LUNS
SELECT REPORT = 01h, ALLOCATION LENGTH = 20h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The REPORT LUNS command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. LUN LIST LENGTH shall be 20h.

3. Data response shall be in Well Known Logical Unit Extended Addressing Format</td></tr>
</table>

7.17 SYNCHRONIZE CACHE (10) Command

The following test cases may be applied to all configured logical units.

7.17.1 Test Case Id: UFS_SynchronizeCache_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.22</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of SYNCHRONIZE CACHE (10) command with SYNC_NV=0 and IMMED = 0b</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue the SYNCHRONIZE CACHE (10) command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>SYNCHRONIZE CACHE (10)
SYNC_NV = 0b, IMMED = 0b, LBA = 00h, GROUP NUMBER = 00h, NUMBER OF LOGICAL BLOCKS = 01h, CONTROL =00h.</td></tr>
<tr><td>Expected Output</td><td>1. The SYNCHRONIZE CACHE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.17.2 Test Case Id: UFS_SynchronizeCache_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.22</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of SYNCHRONIZE CACHE (10) command with SYNC_NV is set to 1</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue the SYNCHRONIZE CACHE (10) command.
2. Verify the Expected Output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>SYNCHRONIZE CACHE (10)
SYNC_NV = 1b, IMMED = 0b, LBA = 00h, GROUP NUMBER = 00h, NUMBER OF LOGICAL BLOCKS = 01h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The SYNCHRONIZE CACHE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.17.3 Test Case Id: UFS_SynchronizeCache_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.22</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of SYNCHRONIZE CACHE (10) command with IMMED is set to 1</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue the SYNCHRONIZE CACHE (10) command.
2. Verify the Expected Output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>SYNCHRONIZE CACHE (10)
SYNC_NV = 0b, IMMED = 1b, LBA = 00h, GROUP NUMBER = 00h, NUMBER OF LOGICAL BLOCKS = 01h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The SYNCHRONIZE CACHE (10) command response shall be:
a) if IMMED is supported
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
b) if IMMED is not supported
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN CDB or NO ADDITIONAL SENSE CODE</td></tr>
</table></td></tr>
</table>

7.17.4 Test Case Id: UFS_SynchronizeCache_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.22</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of SYNCHRONIZE CACHE (10) command with valid LBA field and NUMBER OF LOGICAL BLOCKS set to 0</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue the SYNCHRONIZE CACHE (10) command.
2. Verify the Expected Output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>SYNCHRONIZE CACHE (10)
SYNC_NV = 0b, IMMED = 0b, LBA = 00h, GROUP NUMBER = 00h, NUMBER OF LOGICAL BLOCKS = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The SYNCHRONIZE CACHE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.17.5 Test Case Id: UFS_SynchronizeCache_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.22</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of SYNCHRONIZE CACHE (10) command with invalid LBA field and, NUMBER OF LOGICAL BLOCKS set to 0</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue the READ CAPACITY (10) command and get the Maximum number of logical blocks (Max addressable LBA).
[Main]:
1. Issue the SYNCHRONIZE CACHE (10) command.
2. Verify the Expected Output.
[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>SYNCHRONIZE CACHE (10)
IMMED = 0b, SYNC_NV = 0b, LBA = Maximum addressable LBA + 01h, NUMBER OF LOGICAL BLOCKS = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The SYNCHRONIZE CACHE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE</td></tr>
</table></td></tr>
</table>

7.18 Pre Fetch (10) Command

The following test cases may be applied to all configured logical units

7.18.1 Test Case Id: UFS_PreFetch10_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.19</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of PRE-FETCH (10) command with IMMED is set to 0</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main] .
1. Issue the PRE-FETCH (10) command.
2. Verify the Expected Output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>PRE-FETCH (10)
LBA = 00h, PRE-FETCH LENGTH = 01h, IMMED = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The PRE-FETCH (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
</td></tr>
</table>

7.18.2 Test Case Id: UFS_PreFetch10_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.19</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of PRE-FETCH (10) command with IMMED is set to 1</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue the PRE-FETCH (10) command.
2. Verify the Expected Output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>PRE-FETCH (10)
LBA = 00h, PRE-FETCH LENGTH = 01h, IMMED = 1b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The PRE-FETCH (10) command response shall be:
a) If IMMED is supported
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
b) If IMMED is not supported
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN CDB</td></tr>
</table>
</td></tr>
</table>

7.18.3 Test Case Id: UFS_PreFetch10_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.19</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of PRE-FETCH (10) command with valid LBA field and PRE-FETCH LENGTH set to 0</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue the PRE FETCH (10) command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>PRE-FETCH (10)
LBA = 00h, PRE-FETCH LENGTH = 00h, IMMED = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The PRE-FETCH (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

7.18.4 Test Case Id: UFS_PreFetch10_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.19</td></tr>
<tr><td>Test Purpose</td><td>To verify PRE-FETCH (10) command when LBA exceeds the capacity of the medium.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:

[Main]:
1. Issue READ CAPACITY (10) command to get the MAX addressable LBA (RETURN LOGICAL BLOCK ADDRESS in Parameter Data).
2. Issue the PRE-FETCH (10) command.
3. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>PRE-FETCH (10)
LBA = Max addressable LBA + 01h, PRE-FETCH LENGTH = 00h, IMMED = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The PRE-FETCH (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE</td></tr>
</table></td></tr>
</table>

7.18.5 Test Case Id: UFS_PreFetch10_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.19</td></tr>
<tr><td>Test Purpose</td><td>To verify the status of PRE-FETCH (10) command with valid LBA plus PRE-FETCH LENGTH exceeds the capacity of medium.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue READ CAPACITY (10) command and get the MAX addressable LBA (RETURN LOGICAL BLOCK ADDRESS in Parameter Data).
2. Issue the PRE-FETCH (10) command.
3. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>PRE-FETCH (10)
LBA = Max addressable LBA, PRE-FETCH LENGTH = 02h, IMMED = 0b, GROUP NUMBER = 00h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The PRE-FETCH (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>LOGICAL BLOCK ADDRESS OUT OF RANGE</td></tr>
</table></td></tr>
</table>

8 UFS Protocol Test

8.1 Context Management

8.1.1 In the following test cases, all configured logical units may be used as INDEX to configure the context in the particular LU (INDEX = n).Test Case Id: UFS_ContextManagement_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 13.4.9</td></tr>
<tr><td>Test Purpose</td><td>To verify that context ID configured for Write operation is working for Write commands.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue Query Request Read Descriptor to read the Unit Descriptor of logical unit “N”.
2. Verify that MaxContextID field of wContextCapabilities is not zero.

[Main]:
1. Issue Query Request Write Attribute command to set wContextConf for Write operation.
2. Verify the Expected Output.<1>
3. Issue WRITE (10) command.
4. Verify the Expected Output.<2>

[Clean up]:
1. Issue Query Request Write Attribute command to clear wContextConf (INDEX=N).</td></tr>
<tr><td>Input parameter values</td><td>Query Request: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 10h, INDEX = n,
SELECTOR = 01h, ATTRIBUTE VALUE = 01h.

WRITE (10)
LUN in UPIU = N
LBA = 00h, TRANSFER LENGTH = 01h, WRPROTECT = 000b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUPNUM = 01h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>Attribute IDN</th><th>Query Response</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>wContextConf (10h)</td><td>Success (00h)</td></tr>
</table>
Query Function = 81h, INDEX = n, SELECTOR = 01h, ATTRIBUTE VALUE = 01h

2. The WRITE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

8.1.2 Test Case Id: UFS_ContextManagement_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 13.4.9</td></tr>
<tr><td>Test Purpose</td><td>To verify that context ID configured for read operation is working for Read commands.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue Query Request Read Descriptor to read the Unit Descriptor of logical unit “N”.
2. Verify that MaxContextID field of wContextCapabilities of the LUN is not zero.

[Main]:
1. Issue Query Request Write Attribute command to set wContextConf for read operation.
2. Verify the Expected Output.<1>
3. Issue READ (10) command.
4. Verify the Expected Output.<2>

[Clean up]:
1. Issue Query Request Write Attribute command to clear wContextConf (INDEX = N).</td></tr>
<tr><td>Input parameter values</td><td>Query Request: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 10h, INDEX = n,
SELECTOR = 01h, ATTRIBUTE VALUE = 02h

READ (10)
LUN in UPIU = N
LBA = 00h, TRANSFER LENGTH = 01h, RDPROTECT = 000b, DPO = 0b,
FUA = 0b, FUA_NV = 0b, GROUPNUM = 01h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table><tr><th>OPCODE</th><th>Attribute IDN</th><th>Query Response</th></tr><tr><td>WRITE ATTRIBUTE (04h)</td><td>wContextConf (10h)</td><td>Success (00h)</td></tr></table>Query Function = 81h, INDEX = n, SELECTOR = 01h, ATTRIBUTE VALUE = 02h

2. The READ (10) command response shall be:
<table><tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr><tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr></table></td></tr>
</table>

8.1.3 Test Case Id: UFS_ContextManagement_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 13.4.9</td></tr>
<tr><td>Test Purpose</td><td>To verify that context ID configured for read and write operation is working for Read as well as write commands.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue Query Request Read Descriptor to read the Unit Descriptor of logical unit “N”.
2. Verify that MaxContextID field of wContextCapabilities of the LUN is not zero.

[Main]:
1. Issue Query Request Write Attribute command to set wContextConf for read and write operation.
2. Verify Expected Output.<1>
3. Issue READ (10) command.
4. Verify the Expected Output.<2>
5. Issue WRITE (10) command.
6. Verify the Expected Output.<3>

[Clean up]:
1. Issue Query Request Write Attribute command to clear wContextConf (INDEX = N).</td></tr>
<tr><td>Input parameter values</td><td>Query Request: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 10h, INDEX = n,
SELECTOR = 01h, ATTRIBUTE VALUE = 03h

READ (10)
LUN in UPIU = N
LBA = 00h, TRANSFER LENGTH = 01h, RDPROTECT = 000b, DPO = 0b,
FUA = 0b, FUA_NV = 0b, GROUPNUM = 01h, CONTROL = 00h.

WRITE (10)
LUN in UPIU = N
LBA = 00h, TRANSFER LENGTH = 01h, WRPROTECT = 000b, DPO = 0b,
FUA = 0b, FUA_NV = 0b, GROUPNUM = 01h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table><tr><th>OPCODE</th><th>Attribute IDN</th><th>Query Response</th></tr><tr><td>WRITE ATTRIBUTE (04h)</td><td>wContextConf (10h)</td><td>Success (00h)</td></tr></table>Query Function = 81h, INDEX = n, SELECTOR = 01h, ATTRIBUTE VALUE = 03h

2. The READ (10) command response shall be:
<table><tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr><tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr></table>
3. The WRITE (10) command response shall be:
<table><tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr><tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr></table></td></tr>
</table>

8.1.4 Test Case Id: UFS_ContextManagement_04

Obsolete

8.1.5 Test Case Id: UFS_ContextManagement_05

Obsolete

8.2 UFS Task Management

The following test cases may be applied to all configured logical units.

8.2.1 Test Case ID: UFS_TM_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.7.7.1</td></tr>
<tr><td>Test Purpose</td><td>To verify ABORT TASK function with command queue empty.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Task Management Request (ABORT TASK).
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Task Management Request UPIU: ABORT TASK
Task Management Function = 01h, Input Parameter 1 (LUN) = LUN in UPIU, Input Parameter 2 (Task Tag) = 01h.</td></tr>
<tr><td>Expected Output</td><td>1. Task Management response shall be:
<table>
<tr><th>Response</th><th>Task Management Service Response</th></tr>
<tr><td>Target Success</td><td>Task Management Function Complete (00h)</td></tr>
</table>
</td></tr>
</table>

8.2.2 Test Case ID: UFS_TM_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.7.7.2</td></tr>
<tr><td>Test Purpose</td><td>To Verify ABORT TASK SET function with command queue empty.</td></tr>
<tr><td>Test Procedure</td><td>[Pre-Condition]:
None

[Main]
1. Issue Task Management Request (ABORT TASK SET) .
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Task Management Request UPIU: ABORT TASK SET
Task Management Function = 02h, Input Parameter 1 (LUN) = LUN in UPIU, Input Parameter 2 (Task Tag) = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Task Management response shall be:
<table>
<tr><th>Response</th><th>Task Management Service Response</th></tr>
<tr><td>Target Success</td><td>Task Management Function Complete (00h)</td></tr>
</table></td></tr>
</table>

8.2.3 Test Case ID: UFS_TM_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.7.7.3</td></tr>
<tr><td>Test Purpose</td><td>To verify CLEAR TASK SET function with command queue empty.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Task Management Request (CLEAR TASK SET).
2. Verify the Expected Output. .

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Task Management Request UPIU: CLEAR TASK SET
Task Management Function = 04h, Input Parameter 1 (LUN) = LUN in UPIU.
Input Parameter 2 (Task Tag) = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Task Management response shall be:
<table>
<tr><th>Response</th><th>Task Management Service Response</th></tr>
<tr><td>Target Success</td><td>Task Management Function Complete (00h)</td></tr>
</table></td></tr>
</table>

8.2.4 Test Case ID: UFS_TM_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.7.7.5</td></tr>
<tr><td>Test Purpose</td><td>To verify QUERY TASK function with command queue empty.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Task Management Request (QUERY TASK).
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Task Management Request UPIU: QUERY TASK
Task Management Function = 80h, Input Parameter 1 (LUN) = LUN in UPIU, Input Parameter 2 (Task Tag) = 01h.</td></tr>
<tr><td>Expected Output</td><td>1. Task Management response shall be:
<table>
<tr><th>Response</th><th>Task Management Service Response</th></tr>
<tr><td>Target Success</td><td>Task Management Function Complete (00h)</td></tr>
</table></td></tr>
</table>

8.2.5 Test Case ID: UFS_TM_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.7.7.6</td></tr>
<tr><td>Test Purpose</td><td>To verify QUERY TASK SET function with command queue empty.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Task Management Request (QUERY TASK SET).
2. Verify the Expected Output.

[Cleanup]:
None.</td></tr>
<tr><td>Input parameter values</td><td>Task Management Request UPIU: QUERY TASK SET
Task Management Function = 81h, Input Parameter 1 (LUN) = LUN in UPIU, Input Parameter 2 (Task Tag) = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Task Management response shall be:
<table>
<tr><th>Response</th><th>Task Management Service Response</th></tr>
<tr><td>Target Success</td><td>Task Management Function Complete (00h)</td></tr>
</table></td></tr>
</table>

8.2.6 Test Case ID: UFS_TM_06

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.9</td></tr>
<tr><td>Test Purpose</td><td>To verify Task Management Request with invalid function code returns Service Response as Task Management Function Not Supported.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Task Management Request command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Task Management Request UPIU
Task Management Function = 03h, Input Parameter 1 (LUN) = LUN in UPIU.
Input Parameter 2 (Task Tag) = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Task Management response shall be:
<table>
<tr><th>Response</th><th>Task Management Service Response</th></tr>
<tr><td>Target Failure</td><td>Task Management Function Not Supported (04h)</td></tr>
</table>
</td></tr>
</table>

8.3 UFS BOOT

8.3.1 Test Case ID: UFS_BOOT_01

The following test case may be applied Device supports Boot feature.

Ref. specs Section	UFS: 13.1.3
Test Purpose	To verify BOOT procedure sequence is operated with read of boot code
Test Procedure	**[Precondition]:** 1. Store data in LU assigned Boot well known LU from LBA zero to LBA = 02h **[Main]:** 1. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset) 2. Issue NOP OUT UPIU.<a> 3. If Host does not receive NOP IN UPIU, go to step 2. 4. Verify the Expected Output.<1> If bDescrAccessEn field in Device Descriptor is equal to one then, A. Issue Query Request Read Descriptor command.<b> B. Verify the Expected Output.<2> 5. Issue TEST UNIT READY command <c> 6. Verify the Expected Output.<3> 7. If Boot LU is not ready, go to step 5. 8. Issue READ (10) command.<d> 9. Verify the Expected Output.<4> 10. Issue Query Request Set Flag command.<e> 11. Verify the Expected Output.<5> 12. Issue Query Request Read Flag command.<f> 13. Verify the Expected Output.<6> 14. If fDeviceInit is not zero, go to step 12. **[Clean up]:** None

<table>
<tr><td>Input parameter values</td><td>
a) NOP OUT UPIU
Transaction Type = 000000b, Flags = 00h, Data Segment Length = 0000h

b) Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 00h,
INDEX = 00h, SELECTOR = 00h, LENGTH = -40h.

c) TEST UNIT READY
LUN field in UPIU = B0h,.
CONTROL = 00h

d) READ (10)
LBA = 00h, TRANSFER LENGTH = 03h, RDPROTECT = 0b,
DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.

e) Query Request UPIU: Set Flag
Query Function = 81h, OPCODE = 06h, FLAG IDN = 01h, INDEX = 00h, SELECTOR = 00h.

f) Query Request UPIU: Read Flag
Query Function = 01h, OPCODE = 05h, FLAG IDN = 01h, INDEX = 00h, SELECTOR = 00h.
</td></tr>
<tr><td>Expected Output</td><td>
1. The Device shall return NOP IN UPIU as following:
Transaction Type = 100000b, Flag = 00h, Response = 00h,
Device Information = 00h, Data Segment Length = 0000h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>00h</td><td>Success (00h)</td></tr>
</table>
bBootEnable = 01h

3. The command response shall be:
a. If TEST UNIT READY command is first time:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>UNIT ATTENTION</td><td>POWER ON RESET or BUS DEVICE RESET OCCURRED or POWER ON RESET OCCURRED OR NO ADDITIONAL SENSE CODE</td></tr>
</table>
b. If TEST UNIT READY command is not first time:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
</td></tr>
</table>

Success	or NOT READY		

4. The READ (10) command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. Query response shall be:

OPCODE	FLAG IDN	Query Response	FLAG VALUE
SET FLAG (06h)	01h	Success (00h)	01h

Query Function = 81h, INDEX = 00h, SELECTOR = 00h.

6. Query response shall be:

OPCODE	FLAG IDN	Query Response	FLAG VALUE
READ FLAG (05h)	01h	Success (00h)	00h or 01h

Query Function = 01h, INDEX = 00h, SELECTOR = 00h

8.3.2 Test Case ID: UFS_BOOT_02

The following test case may be applied Device supports Boot feature.

Ref. specs Section	UFS: 13.1.3
Test Purpose	To verify BOOT procedure sequence is operated with only mandatory sequence.
Test Procedure	**[Precondition]:** None **[Main]:** 1. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset) 2. Issue NOP OUT UPIU.<a> 3. If Host does not receive NOP IN UPIU, go to step 2. 4. Verify the Expected Output.<1> 5. Issue Query Request Set Flag command.<b> 6. Verify the Expected Output.<2> 7. Issue Query Request Read Flag command.<c> 8. Verify the Expected Output.<3> 9. If fDeviceInit is not zero, go to step 7. **[Clean up]:** None
Input parameter values	**a) NOP OUT UPIU** Transaction Type = 000000b, Flags = 00h, Data Segment Length = 0000h. **b) Query Request UPIU: Set Flag** Query Function = 81h, OPCODE = 06h, FLAG IDN = 01h, INDEX = 00h, SELECTOR = 00h. **c) Query Request UPIU: Read Flag** Query Function = 01h, OPCODE = 05h, FLAG IDN = 01h, INDEX = 00h, SELECTOR = 00h.

<table>
<tr>
<td>Expected Output</td>
<td>
1. The Device shall return NOP IN UPIU as following:
Transaction Type = 100000b, Flag = 00h, Response = 00h,
Device Information = 00h, Data Segment Length = 0000h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>01h</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h.

3. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>READ FLAG (05h)</td><td>01h</td><td>Success (00h)</td><td>00h or 01h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h
</td>
</tr>
</table>

8.4 UFS Descriptor

TABLE 1 — Configuration Descriptor Header and Device Descriptor Configurable parameters

Offset	Size	Name	Value
00h	1	bLength	90h
01h	1	bDescriptorType	01h
02h	1	bNumberLU	00h
03h	1	bBootEnable	01h
04h	1	bDescrAccessEn	01h
05h	1	bInitPowerMode	01h
06h	1	bHighPriorityLUN	00h
07h	1	bSecureRemovalType	00h
08h	1	bInitActiveICCLevel	00h
09h	2	wPeriodicRTCUpdate	0000h
0Bh:0Fh	5	Reserved	00h

TABLE 2 — Unit Descriptor Configurable parameters

Offset	Size	Name	LUN=0	LUN=1	LUN=2	LUN=3	LUN=4	LUN=5	LUN=6	LUN=7
10h +16*LUN	1	bLUEnable	01h	01h	01h	01h	01h	01h	01h	01h
11h+16*LUN	1	bBootLunID	00h	00h	01h	02h	00h	00h	00h	00h
12h+16*LUN	1	bLUWriteProtect	00h	01h	02h	00h	00h	01h	02h	00h
13h+16*LUN	1	bMemoryType	00h	00h	00h	00h	00h	00h	00h	00h
14h+16*LUN	4	dNumAllocUnits	1000h	1000h	1000h	1000h	1000h	1000h	1000h	1000h
18h+16*LUN	1	bDataReliability	00h	01h	00h	01h	00h	01h	00h	01h
19h+16*LUN	1	bLogicalBlockSize	0Ch	0Ch	0Ch	0Ch	0Ch	0Ch	0Ch	0Ch
1Ah+16*LUN	1	bProvisioningType	00h	02h	03h	00h	00h	02h	03h	00h
1Bh+16*LUN	2	wContextCapabilities	00h	01h	02h	03h	04h	05h	00h	01h
1Dh+16*LUN	3	Reserved	00h	00h	00h	00h	00h	00h	00h	00h

8.4.1 Test Case Id: UFS_QR_ReadDescriptor_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify that Query request Read Descriptor is working for Device Descriptor.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:

1. Issue Query request Read Descriptor command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 00h, INDEX = 00h, SELECTOR = 00h, LENGTH = -40h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>00h</td><td>Success (00h)</td></tr>
</table>
2. Device Descriptor parameters values shall be set according to the UFS specification (no parameter shall be set to a reserved values).</td></tr>
</table>

8.4.2 Test Case Id: UFS_QR_ReadDescriptor_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify that Query Request Read Descriptor is working for Configuration Descriptor.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition] :
None

[Main]:

1. Issue Query Request Read Descriptor command.
2. Verify the Expected Output.

[Cleanup]:
None.</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 01h, INDEX = 00h, SELECTOR = 00h, LENGTH = 90h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>01h</td><td>Success (00h)</td></tr>
</table>
2. Configuration Descriptor parameters values shall be set according to the UFS specification (no parameter shall be set to a reserved values).</td></tr>
</table>

8.4.3 Test Case Id: UFS_QR_ReadDescriptor_03

This test case may apply to all Unit Descriptors and RPMB Descriptor

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify that Query Request Read Descriptor is working for Unit Descriptor.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:

1. Issue Query Request Read Descriptor command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 02h, INDEX = 00h, SELECTOR = 00h, LENGTH = 23h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>02h</td><td>Success (00h)</td></tr>
</table>
2. Descriptor parameters values shall be set according to the UFS specification (no parameter shall be set to a reserved value).</td></tr>
</table>

8.4.4 Test Case Id: UFS_QR_ReadDescriptor_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify that Query Request Read Descriptor is working for Read Interconnect Descriptor.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:

1. Issue Query Request Read Descriptor command.
2. Verify the Expected Output.

[Cleanup] :
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 04h, INDEX = 00h, SELECTOR = 00h, LENGTH = 06h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>04h</td><td>Success (00h)</td></tr>
</table>
2. Descriptor parameters values shall be set according to the UFS specification.</td></tr>
</table>

8.4.5 Test Case Id: UFS_QR_ReadDescriptor_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify that Query Request Read Descriptor is working for String Descriptor Manufacture Name.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition] :

1. Issue Query Request to retrieve the MANUFACTURER NAME index value from the Device Descriptor (N=iManufacturerName).

[Main]:

1. Issue Query Request Read Descriptor command.
2. Verify the Expected Output.
[Cleanup]:

None.</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h DESCRIPTOR IDN = 05h, INDEX = N, SELECTOR = 00h, LENGTH = FEh.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table><tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr><tr><td>READ DESCRIPTOR (01h)</td><td>05h</td><td>Success (00h)</td></tr></table>2. String Descriptor value shall be set according to the UFS specification.</td></tr>
</table>

8.4.6 Test Case Id: UFS_QR_ReadDescriptor_06

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify that Query Request Read Descriptor is working for String Descriptor Product Name.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue Query Request to retrieve the PRODUCT NAME index value from the Device Descriptor (N=iProductName).
[Main]:
1. Issue Query Request Read Descriptor command.
2. Verify the Expected Output
[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 05h, INDEX = N, SELECTOR = 00h, LENGTH = FEh.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>05h</td><td>Success (00h)</td></tr>
</table>
2. String Descriptor value shall be set according to the UFS specification.</td></tr>
</table>

8.4.7 Test Case Id: UFS_QR_ReadDescriptor_07

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify that Query Request Read Descriptor is working for String Descriptor OEM ID.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:

1. Issue Query Request to retrieve the OEM ID index value from the Device Descriptor (N=iOemID) .

[Main]:

1. Issue Query Request Read Descriptor command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 05h, INDEX = N, SELECTOR = 00h, LENGTH = FEh.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>05h</td><td>Success (00h)</td></tr>
</table>
2. String Descriptor value shall be set according to the UFS specification.</td></tr>
</table>

8.4.8 Test Case Id: UFS_QR_ReadDescriptor_08

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify that Query Request Read Descriptor is working for String Descriptor Serial Number.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:

1. Issue Query Request to retrieve the Serial Number index value from the Device Descriptor (N=iSerialNumber) .

[Main]

1. Issue Query Request Read Descriptor command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 05h, INDEX = N, SELECTOR = 00h, LENGTH = FEh.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>05h</td><td>Success (00h)</td></tr>
</table>
2. String Descriptor value shall be set according to the UFS specification.</td></tr>
</table>

8.4.9 Test Case Id: UFS_QR_ReadDescriptor_09

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify that Query Request Read Descriptor is working for Geometry Descriptor.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:

1. Issue Query Request Read Descriptor command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 07h, INDEX = 00h, SELECTOR = 00h, LENGTH = 48h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>07h</td><td>Success (00h)</td></tr>
</table>
2. Descriptor parameters values shall be set according to the UFS specification (no parameter shall be set to a reserved value).</td></tr>
</table>

8.4.10 Test Case Id: UFS_QR_ReadDescriptor_10

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify that Query Request Read Descriptor is working for Power Descriptor.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:

1. Issue Query Request Read Descriptor command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 08h, INDEX = 00h, SELECTOR = 00h, LENGTH = 62h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>08h</td><td>Success (00h)</td></tr>
</table>
2. Descriptor parameters values shall be set according to the UFS specification.</td></tr>
</table>

8.4.11 Test Case Id: UFS_QR_ReadDescriptor_11

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify device returns INVALID IDN status when the Read descriptor command is send with invalid IDN value</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:

1. Issue Query Request Read Descriptor command.
2. Verify the Expected Output.

[Cleanup] :
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = FFh, INDEX = 00h, SELECTOR = 00h, LENGTH = 1Fh.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>FFh</td><td>Invalid IDN (FDh) or General failure (FFh)</td></tr>
</table></td></tr>
</table>

8.4.12 Test Case Id: UFS_QR_ReadDescriptor_12

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify device returns INVALID INDEX status when the Read descriptor command for STRING is send with invalid INDEX value</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:

1. Issue Query Request Read Descriptor to retrieve the index values from the Device Descriptor (N1 = iManufacturerName , N2 = iProductName, N3 = iSerialNumber, N4 = iOemID, N is not equal to N1, N2, N3 and N4).

[Main]:

1. Issue Query Request Read Descriptor command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 05h, INDEX = N, SELECTOR = 00h, LENGTH = FEh.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>READ DESCRIPTOR (01h)</td><td>05h</td><td>Invalid INDEX (FCh) or General failure (FFh)</td></tr>
</table></td></tr>
</table>

8.4.13 Test Case Id: UFS_QR_WriteDescriptor_01

This test case may be applied if bConfigDescrLock is set to 00h.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify Query Request Write Descriptor is changing Configuration Descriptor values.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Query Request UPIU to write the Configuration Descriptor <a>
2. Verify the Expected Output.<1>
3. Issue Query Request UPIU to read the Configuration Descriptor
4. Verify the Expected Output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Write Descriptor
Query Function = 81h, OPCODE = 02h DESCRIPTOR IDN =01h, INDEX = 00h, SELECTOR = 00h, LENGTH = 90h, Configuration Descriptor Header and Device Descriptor Conf. parameters are defined in the table 1 and Unit Descriptor configurable parameters are defined in table 2

b) Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN =01h, INDEX = 00h, SELECTOR = 00h, LENGTH = 90h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table><tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr><tr><td>WRITE DESCRIPTOR (02h)</td><td>01h</td><td>Success (00h)</td></tr></table>
2. Query response shall be:
<table><tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr><tr><td>READ DESCRIPTOR (01h)</td><td>01h</td><td>Success (00h)</td></tr></table>Configuration Descriptor Header and Device Descriptor Conf. parameters are defined in the table 1 and Unit Descriptor configurable parameters are defined in table 2</td></tr>
</table>

8.4.14 Test Case Id: UFS_QR_WriteDescriptor_02

This test case may be applied if bConfigDescrLock is set to 00h.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.1 & 14.1</td></tr>
<tr><td>Test Purpose</td><td>To verify device returns INVALID LENGTH status when the Write Configuration Descriptor command with Invalid Length</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:

None

[Main]:

1. Issue Query Request Write Descriptor command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Write Descriptor
Query Function = 81h, OPCODE = 02h, DESCRIPTOR IDN = 01h, INDEX = 00h, SELECTOR = 00h, LENGTH = 85h. Configuration Descriptor Header and Device Descriptor Conf. parameters are defined in the table 1 and Unit Descriptor configurable parameters are defined in table 2</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>WRITE DESCRIPTOR (02h)</td><td>01h</td><td>Invalid Length (F9h) or General failure (FFh)</td></tr>
</table></td></tr>
</table>

8.4.15 Test Case Id: UFS_QR_WriteDescriptor_03

This test case may be applied if bConfigDescrLock is set to 01h.

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.8 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that device returns the "Parameter already written" in Response Data</td></tr>
<tr><td>Test Procedure</td><td>[Precondition] :
None

[Main]:
1. Issue Query Request Write Descriptor command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Write Descriptor
Query Function = 81h, OPCODE = 02h, DESCRIPTOR IDN = 01h, INDEX = 00h, SELECTOR = 00h, LENGTH = 90h. Configuration Descriptor Header and Device Descriptor Conf. parameters are defined in the table 1 and Unit Descriptor configurable parameters are defined in table 2</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>DESCRIPTOR IDN</th><th>Query Response</th></tr>
<tr><td>WRITE DESCRIPTOR (02h)</td><td>01h</td><td>Parameter already written (F8h)
or General failure (FFh)</td></tr>
</table></td></tr>
</table>

8.5 UFS Flag

8.5.1 Test Case Id: UFS_QR_SetFlag_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.9 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that device returns the "Parameter not writable" in Response Data when tried to set Read only flag (fBusyRTC).</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Set Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Set Flag
Query Function = 81h, OPCODE = 06h, FLAG IDN = 09h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>09h</td><td>Parameter not Writable (F7h) or General Failure (FFh)</td><td>00h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.2 Test Case Id: UFS_QR_SetFlag_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.9 10.5.11.10 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that the fPowerOnWPEn flag can be set only one time per power cycle</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Execute a power cycle
[Main]:
1. Issue Query Request Set Flag command.<a>
2. Verify the Expected Output.<1>
3. Issue Query Request Set Flag command.<c>
4. Verify the Expected Output.<3>
[Cleanup]:
1. Restart the Device.</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Set Flag
Query Function = 81h, OPCODE = 06h, FLAG IDN = 03h, INDEX = 00h, SELECTOR = 00h.

b) Query Request UPIU: Set Flag
Query Function = 81h, OPCODE = 06h, FLAG IDN = 03h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response <a> shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>03h</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h.

2. Query response <c> shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>03h</td><td>Parameter already written (F8h) or General failure (FFh)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h.</td></tr>
</table>

8.5.3 Test Case Id: UFS_QR_SetFlag_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.9 10.5.11.10 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To Verify the Query Request sent with invalid Query function (3Fh).</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Set Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Set Flag
Query Function = 3Fh, OPCODE = 06h, FLAG IDN = 04h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th></tr>
<tr><td>SET FLAG (06h)</td><td>04h</td><td>General failure (FFh)</td></tr>
</table>
Query Function = 3Fh, INDEX = 00h, SELECTOR = 00h.</td></tr>
</table>

8.5.4 Test Case Id: UFS_QR_SetFlag_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.9 10.5.11.10 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that fBackgroundOpsEn flag can be set,.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue Query Request Clear Flag command with FLAG IDN = 04h (fBackgroundOpsEn).
2. Verify that command executed with query response of success and verify that the returned flag value is zero.

[Main]
1. Issue Query Request Set Flag command.
2. Verify the Expected Output.<1>

[Cleanup] :
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Set Flag
Query Function = 81h, OPCODE = 06h, FLAG IDN = 04h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>04h</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.5 Test Case Id: UFS_QR_SetFlag_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.9 10.5.11.10 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify the Query Request Set Flag command sent with Query Function set to 01h (Standard Read Request).</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue Query Request to clear fBackgroundOpsEn flag (FLAG IDN = 04).
2. Verify that request is successfully executed and the returned flag value is zero.

[Main]:
1. Issue Query Request Set Flag command.
2. Verify the Expected Output.

[Cleanup] :
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Set Flag
Query Function = 01h, OPCODE = 06h, FLAG IDN = 04h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>04h</td><td>Invalid OPCODE (FEh)
or
General Failure (FFh)</td><td>00h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.6 Test Case Id: UFS_QR_SetFlag_06

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.8, 10.5.10.9, 10.5.10.10 10.5.11.9, 10.5.11.10, 10.5.11.11 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that "fPurgeEnable" flag can be written multiple times using Set Flag and Clear Flag commands.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. All LU queues should be empty
2. Execute SYNCHRONIZE CACHE

[Main]:
1. Issue Query Request Set Flag command.
2. Verify the Expected Output<1>
3. Issue Query Request Clear Flag command.
4. Verify the Expected Output.<2>

[Cleanup] :
The purge operation shall be resumed by setting fPurgeEnable to one and the host shall wait until the operation completes.</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Set Flag
Query Function = 81h, OPCODE = 06h, FLAG IDN = 06h, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Clear Flag
Query Function = 81h, OPCODE = 07h, FLAG IDN = 06h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>06h</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>CLEAR FLAG (07h)</td><td>06h</td><td>Success (00h)</td><td>00h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

NOTE 1: The duration of a purge operation may long if there is a large number of unmapped LBAs. This test case may be executed earlier than test cases with UNMAP command to reduce the test time.

NOTE 2: If there is no unmapped LBA the duration of the purge operation may be short, therefore the device may clears fPurgeEnable flag before receiving the second Query Request.

8.5.7 Test Case Id: UFS_QR_SetFlag_07

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.8, 10.5.10.9, 10.5.10.10 10.5.11.9, 10.5.11.10, 10.5.11.11 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that Set only flag can only set by Set Flag command and does not get cleared by clear Flag operation on Flag "fDeviceInit".</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Set Flag command.
2. Verify the Expected Output.<1>
3. Issue Query Request Clear Flag command.
4. Verify the Expected Output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Set Flag
Query Function = 81h, OPCODE = 06h, FLAG IDN = 01h, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Clear Flag
Query Function = 81h, OPCODE = 07h, FLAG IDN = 01h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>01h</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h.

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>CLEAR FLAG (07h)</td><td>01h</td><td>Invalid value (FAh) or General Failure (FFh)</td><td>Any</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h.</td></tr>
</table>

8.5.8 Test Case Id: UFS_QR_SetFlag_08

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.8, 10.5.10.9, 10.5.10.10 10.5.11.9, 10.5.11.10, 10.5.11.11 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>Test to verify the multiple write operation on Flag "fPurgeEnable".</td></tr>
<tr><td>Test Procedure</td><td>[Precondition] :
1. All LU queues should be empty
2. Execute SYNCHRONIZE CACHE

[Main]:
1. Issue Query Request Set Flag command.<a>
2. Verify the Expected Output.<1>
3. Issue Query Request Set Flag command.
4. Verify the Expected Output<2>

[Cleanup]:
The host shall wait until the operation completes.</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Set Flag
Query Function = 81h, OPCODE = 06h, FLAG IDN = 06h, INDEX = 00h, SELECTOR = 00h.

b) Query Request UPIU: Set Flag
Query Function = 81h, OPCODE = 06h, FLAG IDN = 06h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response<a> shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>06h</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>06h</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

NOTE 1 The duration of a purge operation may long if there is a large number of unmapped LBAs. This test case may be executed earlier than test cases with UNMAP command to reduce the test time.

8.5.9 Test Case Id: UFS_QR_SetFlag_09

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.10.9 10.5.11.10 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that fPowerOnWPEn flag value is set to zero after a power cycle or hardware reset.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Set Flag command. <a>
2. Verify the Expected Output.<1>
3. Execute a power cycle
4. Perform device initialization.
5. Issue Query Request Read Flag command.
6. Verify the Expected Output.<2>
7. Issue Query Request Set Flag command. <a>
8. Verify the Expected Output.<1>
9. Assert and de-assert hardware reset signal input (RST_n)
10. Perform device initialization.
11. Issue Query Request Read Flag command.
12. Verify the Expected Output.<2>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Set Flag
Query Function = 81h, OPCODE = 06h, FLAG IDN = 03h, INDEX = 00h, SELECTOR = 00h.

b) Query Request UPIU: Read Flag
Query Function = 01h, OPCODE = 05h, FLAG IDN = 03h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>03h</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h.
.
2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>READ FLAG (05h)</td><td>03h</td><td>Success (00h)</td><td>00h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h.</td></tr>
</table>

8.5.10 Test Case Id: UFS_QR_ReadFlag_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.8 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Flag when FLAG IDN = 01h(fDeviceInit) returns the Flag Value in Response Data</td></tr>
<tr><td>Test Procedure</td><td>[Precondition] :
1. Flag value is not set with any setting after reset (Power-on reset, H/W reset, End point reset, Host System UniPro reset).

[Main]:
1. Issue Query Request Read Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Flag
Query Function = 01h, OPCODE = 05h, FLAG IDN = 01h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>READ FLAG (05h)</td><td>01h</td><td>Success (00h)</td><td>00h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.11 Test Case Id: UFS_QR_ReadFlag_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.8 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Flag when FLAG IDN = 02h(fPermanentWPEn) returns the Flag Value in Response Data</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]
1. Issue Query Request Read Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Flag
Query Function = 01h, OPCODE = 05h, FLAG IDN = 02h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>READ FLAG (05h)</td><td>02h</td><td>Success (00h)</td><td>00h or 01h</td></tr>
<tr><td colspan="4">Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table></td></tr>
</table>

8.5.12 Test Case Id: UFS_QR_ReadFlag_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.8 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Flag when FLAG IDN = 03h(fPowerOnWPEn) returns the Flag Value in Response Data</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Flag value is not set with any setting after reset (Power-on reset, H/W reset).

[Main]
1. Issue Query Request Read Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Flag
Query Function = 01h, OPCODE = 05h, FLAG IDN = 03h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>READ FLAG (05h)</td><td>03h</td><td>Success (00h)</td><td>00h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.13 Test Case Id: UFS_QR_ReadFlag_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.8 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Flag when FLAG IDN = 04h(fBackgroundOpsEn) returns the Flag Value in Response Data</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Flag value is not set with any setting after reset (Power-on reset, H/W reset, End point reset and Host System UniPro reset).

[Main]
1. Issue Query Request Read Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Flag
Query Function = 01h, OPCODE = 05h, FLAG IDN = 04h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>READ FLAG (05h)</td><td>04h</td><td>Success (00h)</td><td>1b</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.14 Test Case Id: UFS_QR_ReadFlag_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.8 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Flag Response when FLAG IDN = 06h(fPurgeEnable) returns the Flag Value in Response Data</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Flag
Query Function = 01h, OPCODE = 05h, FLAG IDN = 06h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>READ FLAG (05h)</td><td>06h</td><td>Parameter not readable (F6h)
or
General Failure(FFh)</td><td>00h or 01h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.15 Test Case Id: UFS_QR_ReadFlag_06

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.8, 10.5.11.9 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Flag Response when FLAG IDN = 08h(fPhyResourceRemoval) returns the Flag Value in Response Data</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Bit 0 value in the wExceptionEventStatus shall be zero.

[Main]:
1. Issue Query Request Read Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Flag
Query Function = 01h, OPCODE = 05h, FLAG IDN = 08h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>READ FLAG (05h)</td><td>08h</td><td>Success (00h)</td><td>0b</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.16 Test Case Id: UFS_QR_ReadFlag_07

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.8 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Flag response when FLAG IDN = 09h(fBusyRTC) returns the Flag Value in Response Data</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Flag
Query Function = 01h, OPCODE = 05h, FLAG IDN = 09h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>READ FLAG (05h)</td><td>09h</td><td>Success (00h)</td><td>0b</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.17 Test Case Id: UFS_QR_ReadFlag_08

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.8 10.5.11.9 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To Verify the Query Request Read Flag command sent with Query Function set to 81h.(Standard Write Request)</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Flag
Query Function = 81h, OPCODE = 05h, FLAG IDN = 04h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG Value</th></tr>
<tr><td>READ FLAG (05h)</td><td>04h</td><td>Invalid OPCODE (FEh) or General failure (FFh)</td><td>0b or 1b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.18 Test Case Id: UFS_QR_ClearFlag_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.10 10.5.11.11 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>Test to verify the Clear flag operation on the fBackgroundOpsEn.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Set Flag command.
2. Verify the Expected Output.<1>
3. Issue Query Request Clear Flag command.
4. Verify the Expected Output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>QUERY REQUEST UPIU: SET FLAG
Query Function = 81h, OPCODE = 06h, FLAG IDN = 04h, INDEX = 00h, SELECTOR = 00h.

QUERY REQUEST UPIU: CLEAR FLAG
Query Function = 81h, OPCODE = 07h, FLAG IDN = 04h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>04h</td><td>Success (00h)</td><td>1b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>CLEAR FLAG (07h)</td><td>04h</td><td>Success (00h)</td><td>0b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.19 Test Case Id: UFS_QR_ClearFlag_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.10 10.5.11.11 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>Test to verify the Clear flag command response when issued for the flag "fPowerOnWPEn" (Read /Power On Reset) once set by Set Flag command.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Flag value is not set with any setting after reset (Power-on reset, H/W reset).

[Main]:
1. Issue Query Request Set Flag command.
2. Verify the Expected Output.<1>
3. Issue Query Request Clear Flag command.
4. Verify the Expected Output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>QUERY REQUEST UPIU: SET FLAG
Query Function = 81h, OPCODE = 06h, FLAG IDN = 03h, INDEX = 00h, SELECTOR = 00h.

QUERY REQUEST UPIU: CLEAR FLAG
Query Function = 81h, OPCODE = 07h, FLAG IDN = 03h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>03h</td><td>Success (00h)</td><td>1b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>CLEAR FLAG (07h)</td><td>03h</td><td>parameter already written (F8h) or General Failure (FFh)</td><td>1b or 0b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.20 Test Case Id: UFS_QR_ClearFlag_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.10 10.5.11.11 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that device returns the "Parameter not writable" in Response to Clear flag command for flag “fBusyRTC".</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Clear Flag command.
2. Verify the Expected Output.<1>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>QUERY REQUEST UPIU: CLEAR FLAG
Query Function = 81h, OPCODE = 07h, FLAG IDN = 09h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>CLEAR FLAG (07h)</td><td>09h</td><td>parameter not writable (F7h) or General Failure (FFh)</td><td>0b or 1b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.21 Test Case Id: UFS_QR_ClearFlag_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.10 10.5.11.11 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that the Host cannot reset the "fPhyResourceRemoval" flag by using Clear Flag command.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Bit 0 value in the wExceptionEventStatus shall be zero.

[Main]:
1. Issue Query Request Set Flag command.
2. Verify the Expected Output.<1>
3. Issue Query Request Clear Flag command.
4. Verify the Expected Output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>QUERY REQUEST UPIU: SET FLAG
Query Function = 81h, OPCODE = 06h, FLAG IDN = 08h, INDEX = 00h, SELECTOR = 00h.

QUERY REQUEST UPIU: CLEAR FLAG
Query Function = 81h, OPCODE = 07h, FLAG IDN = 08h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>08h</td><td>Success (00h)</td><td>1b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>CLEAR FLAG (07h)</td><td>08h</td><td>General Failure (FFh) or Invalid Value (FAh)</td><td>0b or 1b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.22 Test Case Id: UFS_QR_ToggleFlag_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.11 10.5.11.12 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>Test to verify the Toggle flag operation to toggle "fBackgroundOpsEn” flag value from 1b to 0b.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Set Flag command.
2. Verify the Expected Output.<1>
3. Issue Query Request Toggle Flag command.
4. Verify the Expected Output.<2>

[Cleanup] :
None</td></tr>
<tr><td>Input parameter values</td><td>QUERY REQUEST UPIU: SET FLAG
Query Function = 81h, OPCODE = 06h, FLAG IDN = 04h, INDEX = 00h, SELECTOR = 00h.

QUERY REQUEST UPIU: TOGGLE FLAG
Query Function = 81h, OPCODE = 08h, FLAG IDN = 04h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>SET FLAG (06h)</td><td>04h</td><td>Success (00h)</td><td>1b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>TOGGLE FLAG (08h)</td><td>04h</td><td>Success (00h)</td><td>0b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.23 Test Case Id: UFS_QR_ToggleFlag_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.11 10.5.11.12 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>Test to verify the Toggle flag command response when issued twice for the flag "fPowerOnWPEn" (Read /Power On Reset)</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Flag value is not set with any setting after reset (Power-on reset, H/W reset).

[Main]:
1. Issue Query Request Toggle Flag command<a>.
2. Verify the Expected Output.<1>
3. Issue Query Request Toggle Flag command
4. Verify the Expected Output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) QUERY REQUEST UPIU: TOGGLE FLAG
Query Function = 81h, OPCODE = 08h, FLAG IDN = 03h, INDEX = 00h, SELECTOR = 00h.

b) QUERY REQUEST UPIU: TOGGLE FLAG
Query Function = 81h, OPCODE = 08h, FLAG IDN = 03h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>TOGGLE FLAG (08h)</td><td>03h</td><td>Success (00h)</td><td>1b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>TOGGLE FLAG (08h)</td><td>03h</td><td>parameter already written (F8h) or General Failure (FFh)</td><td>0b or 1b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.24 Test Case Id: UFS_QR_ToggleFlag_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.11 10.5.11.12 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that device returns the "Parameter not writable" in Response to Toggle Flag command for "fBusyRTC" flag.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Toggle Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>QUERY REQUEST UPIU: TOGGLE FLAG
Query Function = 81h, OPCODE = 08h, FLAG IDN = 09h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FALG VALUE</th></tr>
<tr><td>TOGGLE FLAG (08h)</td><td>09h</td><td>parameter not Writable (F7h) or General Failure(FFh)</td><td>0b or 1b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.5.25 Test Case Id: UFS_QR_ToggleFlag_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: 10.5.10.11 10.5.11.12 and 14.2</td></tr>
<tr><td>Test Purpose</td><td>Test to verify the Toggle flag command response on "fPurgeEnable" flag.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Toggle Flag command.
2. Verify the Expected Output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>QUERY REQUEST UPIU: TOGGLE FLAG
Query Function = 81h, OPCODE = 08h, FLAG IDN = 06h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>FLAG IDN</th><th>Query Response</th><th>FLAG VALUE</th></tr>
<tr><td>TOGGLE FLAG (08h)</td><td>06h</td><td>Success (00h)</td><td>0b or 1b</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6 UFS Attribute

8.6.1 Test Case Id: UFS_QR_ReadAttribute_01

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute command issued with Attribute IDN = 00h (bBootLunEn) returns Success in response.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.
[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 00h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>00h</td><td>Success (00h)</td><td>00h or 01h or 02h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.2 Test Case Id: UFS_QR_ReadAttribute_02

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify that Attribute bCurrentPowerMode is set to valid value according to bInitPowerMode.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset)
[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.
[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 02h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>02h</td><td>Success (00h)</td><td>00h or 11h or 20h or 22h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h

NOTE bCurrentPowerMode value after device initialization is 22h(UFS-Sleep mode) if bInitPowerMode = 00h, or 11h(Active mode) if bInitPowerMode = 01h.</td></tr>
</table>

8.6.3 Test Case Id: UFS_QR_ReadAttribute_03

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify that Attribute bActiveICCLevel is set to bInitActiveICCLevel at Power On.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue the Query Request Read Descriptor command to read the Device Descriptor parameter bInitActiveICCLevel “N”. (00h ≤ N ≤ 0Fh)
2. Attribute value is not set with any setting after reset (Power-on reset, H/W reset)
[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.
[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 03h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>03h</td><td>Success (00h)</td><td>N</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.4 Test Case Id: UFS_QR_ReadAttribute_04

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute command issued with Attribute IDN = 04h (bOutOfOrderDataEn) returns Success in response.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read command.
2. Verify the expected output.
[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 04h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>04h</td><td>Success (00h)</td><td>00h or 01h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.5 Test Case Id: UFS_QR_ReadAttribute_05

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute command issued with Attribute IDN = 05h (bBackgroundOpStatus) returns Success in response.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 05h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>05h</td><td>Success (00h)</td><td>00h, 01h, 02h, or 03h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.6 Test Case Id: UFS_QR_ReadAttribute_06

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute command when Attribute IDN = 06h (bPurgeStatus) returns the Value in Response Data</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset)

[Main]:
1. Issue Query Request with Read Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 06h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>06h</td><td>Success (00h)</td><td>00h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.7 Test Case Id: UFS_QR_ReadAttribute_07

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute command issued with Attribute IDN = 07h (bMaxDataInSize) returns Success in response.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 07h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>07h</td><td>Success (00h)</td><td>Any value</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h

NOTE bMaxDataInSize shall not exceed the bMaxInBuffer Size parameter.</td></tr>
</table>

8.6.8 Test Case Id: UFS_QR_ReadAttribute_08

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the ReadAttribute command issued with Attribute IDN = 08h (bMaxDataOutSize) returns Success in response.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Attribute command..
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 08h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>08h</td><td>Success (00h)</td><td>Any value</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h

NOTE bMaxDataOutSize shall not exceed the bMaxOutBuffer Size parameter.</td></tr>
</table>

8.6.9 Test Case Id: UFS_QR_ReadAttribute_09

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute command when Attribute IDN = 09h (dDynCapNeeded) returns the value in Response Data</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 09h, INDEX = N (N = Configured Logical Unit Number), SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>09h</td><td>Success (00h)</td><td>Any value</td></tr>
</table>
Query Function = 01h, INDEX = N, SELECTOR = 00h</td></tr>
</table>

8.6.10 Test Case Id: UFS_QR_ReadAttribute_10

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute command issued with Attribute IDN = 0Ah (bRefClkFreq) returns Success in response.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Read Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 0Ah, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>0Ah</td><td>Success (00h)</td><td>00h or 01h or 02h or 03h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.11 Test Case Id: UFS_QR_ReadAttribute_11

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute command issued with Attribute IDN = 0Bh (bConfigDescrLock) returns Success in response.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 0Bh, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>0Bh</td><td>Success (00h)</td><td>00h or 01h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.12 Test Case Id: UFS_QR_ReadAttribute_12

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute Command response with Attribute IDN = 0Ch (bMaxNumOfRTT) returns Success in response.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 0Ch, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>0Ch</td><td>Success (00h)</td><td>N (1 < N < bDeviceRTTCap+1)</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h

NOTE bMaxNumOfRTT shall not exceed the bDeviceRTTCap parameter.</td></tr>
</table>

8.6.13 Test Case Id: UFS_QR_ReadAttribute_13

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute Command with Attribute IDN = 0Dh (wExceptionEventControl) returns Success in response.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Attribute value is not set with any setting after reset (Power-on reset, H/W reset, End point reset and Host system UniPro reset)

[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 0Dh, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>0Dh</td><td>Success (00h)</td><td>0000h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.14 Test Case Id: UFS_QR_ReadAttribute_14

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute Command response with Attribute IDN = 0Eh (wExceptionEventStatus) returns Success in response.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]
1. Issue Query Request Read Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 0Eh, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>0Eh</td><td>Success (00h)</td><td>0000h ~ 0007h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.15 Test Case Id: UFS_QR_ReadAttribute_15

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute Command response with Attribute IDN = 0Fh (dSecondsPassed).</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 0Fh, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>0Fh</td><td>Parameter not readable (F6h) or General failure (FFh)</td><td>Any Value</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.16 Test Case Id: UFS_QR_ReadAttribute_16

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute Command response with Attribute IDN = 10h (wContextConf) returns Success in response.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Attribute value is not set with any setting after reset (Power-on reset, H/W reset, End point reset and Host system UniPro reset)

[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 10h, INDEX = N (N = Configured Logical Unit Number), SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>10h</td><td>Success (00h)</td><td>00h</td></tr>
</table>
Query Function = 01h, INDEX = N, SELECTOR = 00h</td></tr>
</table>

8.6.17 Test Case Id: UFS_QR_ReadAttribute_17

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.11.7 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Read Attribute Command response with Attribute IDN = 11h (dCorrPrgBlkNum).</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 11h,
INDEX = N (N = Configured Logical Unit Number), SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>11h</td><td>Success (00h)</td><td>Any Value</td></tr>
</table>
Query Function = 01h, INDEX = N, SELECTOR = 00h</td></tr>
</table>

8.6.18 Test Case Id: UFS_QR_WriteAttribute_01

Ref. specs Section	Section 10.5.10.7, 10.5.11.8 and 14.3
Test Purpose	To verify the bBootLunEn attribute value can be changed multiple times using Write Attribute command.
Test Procedure	**[Precondition]:** None. **[Main]:** 1. Issue Query Request Read Attribute command for getting the initial value of bBootLunEn Attribute.<a> 2. Issue Query Request Write Attribute command.<b> 3. Verify the expected output.<1> 4. Issue Query Request Write Attribute command.<c> 5. Verify the expected output.<2> 6. Issue Query Request Write Attribute command.<d> 7. Verify the expected output.<3> **[Clean up]:** 1. Issue Query Request Write Attribute command to restore bBootLunEn to its Initial value read at step 1.
Input parameter values	**a) Query Request UPIU: Read Attribute** Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 00h, INDEX = 00h, SELECTOR = 00h. **b) Query Request UPIU: Write Attribute** Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 00h, INDEX = 00h, SELECTOR = 00h, Attribute Value = 00h. **c) Query Request UPIU: Write Attribute** Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 00h, INDEX = 00h, SELECTOR = 00h, Attribute Value = 01h. **d) Query Request UPIU: Write Attribute** Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 00h, INDEX = 00h, SELECTOR = 00h, Attribute Value = 02h.

Expected Output

1. Query response shall be:

OPCODE	ATTRIBUTE IDN	Query Response	ATTRIBUTE VALUE
WRITE ATTRIBUTE (04h)	00h	Success (00h)	00h

Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:

OPCODE	ATTRIBUTE IDN	Query Response	ATTRIBUTE VALUE
WRITE ATTRIBUTE (04h)	00h	Success (00h)	01h

Query Function = 81h, INDEX = 00h, SELECTOR = 00h

3. Query response shall be:

OPCODE	ATTRIBUTE IDN	Query Response	ATTRIBUTE VALUE
WRITE ATTRIBUTE (04h)	00h	Success (00h)	02h

Query Function = 81h, INDEX = 00h, SELECTOR = 00h

8.6.19 Test Case Id: UFS_QR_WriteAttribute_02

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Write Attribute command response when issued for the Attribute "bBootLunEn" to set its value to FFh (Invalid).</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]
1. Issue Query Request Read Attribute command for getting initial value “N”
2. Issue Query Request Write Attribute command.
3. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 81h, OPCODE = 03h ATTRIBUTE IDN = 00h, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 00h, INDEX = 00h, SELECTOR = 00h, Attribute Value = FFh.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>00h</td><td>Invalid value (FAh) or General failure (FFh)</td><td>-FFh</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.20 Test Case Id: UFS_QR_WriteAttribute_03

The following test case may be applied when bBootEnable is set to 01h and “Boot LU A” and “Boot LU B” are assigned to configured logical unit.

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the attribute bBootLunEn is of persistent type.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Attribute command to get the initial value of bBootLunEn Attribute.<a>
2. Issue Query Request Write Attribute command.
3. Verify the expected output.<1>
4. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset)
5. Issue Query Request Read Attribute command.<c>
6. Verify the expected output.<2>

[Clean up]:
1. Issue Query Request Write Attribute command to restore bBootLunEn to its Initial value read at step 1.</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 00h, INDEX = 00h, SELECTOR = 00h.

b) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 00h, INDEX = 00h, SELECTOR = 00h, Attribute Value = 01h.

c) Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 00h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>00h</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>00h</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.21 Test Case Id: UFS_QR_WriteAttribute_04

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.6, 10.5.10.7, 10.5.11.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the response of Write Attribute command when issued for the Attribute "bCurrentPowerMode" (Read Only).</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Attribute command for getting initial value “N”
2. Issue Query Request Write Attribute command.
3. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 02h, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 02h, INDEX = 00h, SELECTOR = 00h, Attribute Value = 01h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>02h</td><td>Parameter not writeable (F7h) or General failure (FFh)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.22 Test Case Id: UFS_QR_WriteAttribute_05

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the bActiveICCLevel attribute value can be changed multiple times using Write Attribute command.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Attribute command to get the initial value of bActiveICCLevel Attribute “N”.<a>
2. Issue Query Request Write Attribute command.
3. Verify the expected output.<1>
4. Issue Query Request Write Attribute command.<c>
5. Verify the expected output.<2>

[Cleanup]:
1. Issue Query Request Write Attribute command to restore bActiveICCLevel to its Initial value read at step 1.</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 03h, INDEX = 00h, SELECTOR = 00h.

b) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 03h, INDEX = 00h, SELECTOR = 00h, Attribute Value = N1 (N1 != N).

c) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 03h, INDEX = 00h, SELECTOR = 00h, Attribute Value = N2 (N2 != N1)</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>03h</td><td>Success (00h)</td><td>N1</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>03h</td><td>Success (00h)</td><td>N2</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.23 Test Case Id: UFS_QR_WriteAttribute_06

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Write Attribute command response when issued for the Attribute "bActiveICCLevel" (Read /Persistent) to set its value to 10h (out of range).</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Attribute command for getting initial value “N”
2. Issue Query Request Write Attribute command.
3. Verify the expected output.

[Cleanup] :
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 03h, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 03h, INDEX = 00h, SELECTOR = 00h, Attribute Value = 10h</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>03h</td><td>Invalid Value (FAh) or General failure (FFh)</td><td>10h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.24 Test Case Id: UFS_QR_WriteAttribute_07

This test case may be applied only if bOutOfOrderDataEn attribute has not been written yet.

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Response of Write Attribute command when issued for the attribute "bOutOfOrderDataEn" (Read/Write once) and this attribute has Write Once Access Property.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Query Request Write Attribute command.<a>
2. Verify the expected output.<1>
3. Issue Query Request Write Attribute command
4. Verify the expected output.<2>

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 04h, INDEX = 00h, SELECTOR = 00h, Attribute Value = 01h.

b) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 04h, INDEX = 00h, SELECTOR = 00h, Attribute Value = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>04h</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>04h</td><td>Parameter already written (F8h) or General failure (FFh)</td><td>00h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.25 Test Case Id: UFS_QR_WriteAttribute_08

This test case may be applied only if bOutOfOrderDataEn attribute has not been written yet.

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Write Attribute command returns Invalid Value as response when issued for "bOutOfOrderDataEn" (Read /Write once) with value FFh. (invalid)</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Query Request Read Attribute command for getting initial value “N”
2. Issue Query Request Write Attribute command.
3. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 04h, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 04h, INDEX = 00h, SELECTOR = 00h, Attribute Value = FFh.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>04h</td><td>Invalid value(FAh) or General failure (FFh)</td><td>FFh</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.26 Test Case Id: UFS_QR_WriteAttribute_09

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the device returns the "Parameter not writable" in Response to Write Attribute command for read only attribute "bBackgroundOpStatus".</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Write Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 05h, INDEX = 00h, SELECTOR = 00h, Attribute Value = 01h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>05h</td><td>Parameter not writable (F7h) or General failure (FFh)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.27 Test Case Id: UFS_QR_WriteAttribute_10

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the device returns the "Parameter not writable" in Response to Write Attribute command for "bPurgeStatus" attribute.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request with Write Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 06h, INDEX = 00h, SELECTOR = 00h, Attribute Value = 01h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>06h</td><td>Parameter not writable (F7h) or General failure (FFh)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.28 Test Case Id: UFS_QR_WriteAttribute_11

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the bMaxDataInSize(Read/ Persistent) attribute value can be changed multiple times using Write Attribute command.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Command Queues are empty

[Main]:
1. Issue Query Request Read Attribute command to get the initial value of bMaxDataInSize Attribute.<a>
2. Issue Query Request Write Attribute command.
3. Verify the expected output.<1>
4. Issue Query Request Write Attribute command.<c>
5. Verify the expected output.<2>

[Clean up]:
1. Issue Query Request Write Attribute command to restore bMaxDataInSize to its Initial value read at step 1.</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h ATTRIBUTE IDN = 07h, INDEX = 00h, SELECTOR = 00h.

b) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 07h, INDEX = 00h, SELECTOR = 00h. Attribute Value = N1 (‘N1’ can be any value except with Attribute Value in the response of Query Request Read Attribute command)

c) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 07h, INDEX = 00h, SELECTOR = 00h. Attribute Value = N2 (‘N2’ can be any value except with Attribute Value in the response of Query Request Write Attribute command<a>)

NOTE bMaxDataInSize shall not exceed the bMaxInBufferSize parameter.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>07h</td><td>Success (00h)</td><td>N1</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>07h</td><td>Success (00h)</td><td>N2</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.29 Test Case Id: UFS_QR_WriteAttribute_12

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify that the attribute bMaxDataInSize is of persistent type.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]: .
1. Command Queues are empty

[Main]:
1. Issue Query Request Read Attribute command to get the initial value of bMaxDataInSize Attribute.<a>
2. Issue Query Request Write Attribute command.
3. Verify the expected output.<1>
4. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset)
5. Issue Query Request Read Attribute command.<c>
6. Verify the expected output.<2>

[Clean up]:
1. Issue Query Request Write Attribute command to restore bMaxDataInSize to its Initial value read at Step 1.</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 07h, INDEX = 00h, SELECTOR = 00h.

b) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 07h, INDEX = 00h, SELECTOR = 00h, Attribute Value = N.(‘N’ can be any value except with Attribute Value in the response of Query Request Read Attribute command)

c) Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 07h, INDEX = 00h, SELECTOR = 00h.

NOTE bMaxDataInSize shall not exceed the bMaxInBufferSize parameter.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>07h</td><td>Success (00h)</td><td>N</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>07h</td><td>Success (00h)</td><td>N</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00</td></tr>
</table>

8.6.30 Test Case Id: UFS_QR_WriteAttribute_13

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the response of the Write Attribute command when the attribute value of bMaxDataInSize attribute is tried to set to a value greater than bMaxInBufferSize in Geometry descriptor.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Descriptor command to get value of bMaxInBufferSize.
2. Issue Query Request Read Attribute command for getting initial value “N”
3. Issue Query Request Write Attribute command.
4. Verify the expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 07h, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 07h, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 07h, INDEX = 00h, SELECTOR = 00h, Attribute Value = (bMaxInBufferSize + 01h).</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>07h</td><td>Invalid value(FAh) or General failure (FFh)</td><td>bMaxInBufferSize + 01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.31 Test Case Id: UFS_QR_WriteAttribute_14

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the bMaxDataOutSize(Read/ Persistent) attribute value can be changed multiple times using Write Attribute commands</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Command Queues are empty
[Main]:
1. Issue Query Request Read Attribute command to get the initial value of bMaxDataOutSize Attribute.<a>
2. Issue Query Request Write Attribute command.
3. Verify the expected output.<1>
4. Issue Query Request Write Attribute command.<c>
5. Verify the expected output.<2>
[Clean up]:
1. Issue Query Request Write Attribute command to restore bMaxDataOutSize to its Initial value read at step 1.</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h ATTRIBUTE IDN = 08h, INDEX = 00h, SELECTOR = 00h.

b) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 08h, INDEX = 00h, SELECTOR = 00h. Attribute Value = N1 (‘N1’ can be any value except with Attribute Value in the response of Query Request Read Attribute command)

c) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 08h, INDEX = 00h, SELECTOR = 00h. Attribute Value = N2 (‘N2’ can be any value except with Attribute Value in the response of Query Request Write Attribute command<a>)

NOTE bMaxDataOutSize shall not exceed the bMaxOutBuffer Size parameter.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>08h</td><td>Success (00h)</td><td>N1</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>08h</td><td>Success (00h)</td><td>N2</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.32 Test Case Id: UFS_QR_WriteAttribute_15

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify that the attribute bMaxDataOutSize is of persistent type.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Command Queue are empty

[Main]:
1. Issue Query Request Read Attribute command to get the initial value of bMaxDataOutSize Attribute.<a>
2. Issue Query Request Write Attribute command.
3. Verify the expected output.<1>
4. Reset the device (Power-on reset, H/W reset, End point reset and Host system UniPro reset)
5. Issue Query Request Read Attribute command.<c>
6. Verify the expected output.<2>

[Clean up]:
1. Issue Query Request Write Attribute command to restore bMaxDataOutSize to its initial value read at Step 1.</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 08h, INDEX = 00h, SELECTOR = 00h.

b) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 08h, INDEX = 00h, SELECTOR = 00h, Attribute Value = N (‘N’ can be any value except with Attribute Value in the response of Query Request Read Attribute command)

c) Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 08h, INDEX = 00h, SELECTOR = 00h.

NOTE bMaxDataOutSize shall not exceed the bMaxOutBuffer Size parameter.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>08h</td><td>Success (00h)</td><td>N</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>08h</td><td>Success (00h)</td><td>N</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00</td></tr>
</table>

8.6.33 Test Case Id: UFS_QR_WriteAttribute_16

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the response of the Write Attribute command when the attribute value of bMaxDataOutSize attribute is tried to set to a value greater than bMaxOutBufferSize in Geometry descriptor.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Descriptor command to get value of bMaxOutBufferSize.
2. Issue Query Request Read Attribute command for getting initial value “N”
3. Issue Query Request Write Attribute command.
4. Verify the expected output.

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Descriptor
Query Function = 01h, OPCODE = 01h, DESCRIPTOR IDN = 08h, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 08h, INDEX = 00h, SELECTOR = 00h, Attribute Value = (bMaxOutBufferSize + 01h).</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>08h</td><td>Invalid value(FAh) or General failure (FFh)</td><td>bMaxOutBufferSize + 01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.34 Test Case Id: UFS_QR_WriteAttribute_17

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the device returns the "Parameter not writable" in Response to Write Attribute command for "dDynCapNeeded" attribute.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Write Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 09h, INDEX = N (N = Configured Logical Unit Number), SELECTOR = 00h, Attribute Value = 01h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>09h</td><td>Parameter not writable (F7h) or General failure (FFh)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = N, SELECTOR = 00h</td></tr>
</table>

8.6.35 Test Case Id: UFS_QR_WriteAttribute_18

8.6.36 Test Case Id: UFS_QR_WriteAttribute_19

This test case may be applied only if bRefClkFreq attribute has not been written yet.

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Write Attribute command returns Invalid Value as response when issued for "bRefClkFreq" (Read / Persistent) with value FCh. (invalid)</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Attribute command for getting initial value “N”
2. Issue Query Request Write Attribute command.
3. Verify the expected output.

[Cleanup]:
None.</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 0Ah, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 0Ah, INDEX = 00h, SELECTOR = 00h, Attribute Value = FCh.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>0Ah</td><td>Invalid value(FAh) or General failure (FFh)</td><td>FCh</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.37 Test Case Id: UFS_QR_WriteAttribute_20

This test case may be applied only if bConfigDescrLock attribute has not been written yet.

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Response of Write Attribute command when issued for the attribute "bConfigDescrLock" (Read/Write once) and this attribute has Write Once Access Property.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition] :
None

[Main]:
1. Issue Query Request Write Attribute.<a>
2. Verify the expected output.<1>
3. Issue Query Request Write Attribute.
4. Verify the expected output.<2>

[Cleanup] :
None</td></tr>
<tr><td>Input parameter values</td><td>a) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 0Bh, INDEX = 00h, SELECTOR = 00h, Attribute Value = 01h

b) Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 0Bh, INDEX = 00h, SELECTOR = 00h, Attribute Value = 01h</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>0Bh</td><td>Success (00h)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h

2. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>0Bh</td><td>Parameter already written (F8h) or General failure (FFh)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.38 Test Case Id: UFS_QR_WriteAttribute_21

This test case may be applied only if bConfigDescrLock attribute has not been written yet.

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Write Attribute command returns Invalid value as response when issued for "bConfigDescrLock" (Read /Write once) with value FFh (invalid).</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Read Attribute command for getting initial value 00h
2. Issue Query Request Write Attribute command.
3. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 0Bh, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 0Bh, INDEX = 00h, SELECTOR = 00h, Attribute Value = FFh</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>0Bh</td><td>Invalid value(FAh) or General failure (FFh)</td><td>FFh</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.39 Test Case Id: UFS_QR_WriteAttribute_22

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the wExceptionEventControl attribute value can be changed using Write Attribute command.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Query Request Read Attribute command to get the initial value of wExceptionEventControl Attribute”N”.
2. Issue Query Request Write Attribute command.
3. Verify the expected output.

[Cleanup]:
1. Issue Query Request Write Attribute command to restore wExceptionEventControl to its Initial value read at step 1.</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 0Dh, INDEX = 00h, SELECTOR = 00h.

Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 0Dh, INDEX = 00h, SELECTOR = 00h, Attribute Value = N1(N1 is one of value from 00h to 07h and N1 != N).</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>0Dh</td><td>Success (00h)</td><td>N1</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.40 Test Case Id: UFS_QR_WriteAttribute_23

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the device returns the "Parameter not writable" in Response to Write Attribute command for wExceptionEventStatus (Read only) attribute.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition] :
None.

[Main]:
1. Issue Query Request Write Attribute command.
2. Verify the expected output.

[Cleanup] :
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 0Eh, INDEX = 00h, SELECTOR = 00h, Attribute Value = FFFFh.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>0Eh</td><td>Parameter not writable (F7h) or General failure (FFh)</td><td>FFFFh</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.41 Test Case Id: UFS_QR_WriteAttribute_24

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the Write Attribute command response on "dSecondsPassed".</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Write Attribute command.
2. Verify the expected output.

[Cleanup] :
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h ATTRIBUTE IDN = 0Fh, INDEX = 00h, SELECTOR = 00h, Attribute Value = FFFFh.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>0Fh</td><td>Success (00h)</td><td>FFFFh</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.6.42 Test Case Id: UFS_QR_WriteAttribute_25

This test case may be applied when all contexts are closed.

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7, 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the “wContextConf” attribute value can be changed using Write Attribute command.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition] :
1. Issue Query Request Read Descriptor to read the Unit Descriptor of logical unit “N”.
2. Verify that MaxContextID field of wContextCapabilities is not zero.
[Main]
1. Issue Query Request Write Attribute command.
2. Verify the expected output.
[Cleanup] :
1. Issue Query Request Write Attribute command to close all contexts.</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = 10h, INDEX = N (N = Configured Logical Unit Number), SELECTOR = N1(N1 = supported context ID in LUN N), Attribute Value = 0001h</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>10h</td><td>Success (00h)</td><td>0001h</td></tr>
</table>
Query Function = 81h, INDEX = N, SELECTOR = N1</td></tr>
</table>

8.6.43 Test Case Id: UFS_QR_WriteAttribute_27

<table>
<tr><td>Ref. specs Section</td><td>Section 10.5.10.7 10.5.11.8 and 14.3</td></tr>
<tr><td>Test Purpose</td><td>To verify the device returns invalid IDN status when invalid IDN value is given to Write Attribute command</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None.

[Main]:
1. Issue Query Request Write Attribute command.
2. Verify the expected output.

[Cleanup]:
None</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Write Attribute
Query Function = 81h, OPCODE = 04h, ATTRIBUTE IDN = FFh, INDEX = 00h, SELECTOR = 00h, Attribute Value = 01h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>WRITE ATTRIBUTE (04h)</td><td>FFh</td><td>Invalid IDN (FDh) or General failure (FFh)</td><td>01h</td></tr>
</table>
Query Function = 81h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.7 UFS Power Mode

8.7.1 Test Case ID: UFS_PowerMode_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 7.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that device terminates START STOP UNIT command with CHECK CONDITION status if the POWER CONDITION field value is invalid while it is in UFS Sleep Mode</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Sleep Mode (POWER CONDITION = 02h)
2. Issue Query Request Read Attribute to verify bCurrentPowerMode is set to UFS-Sleep mode.

[Main]:
1. Issue START STOP UNIT command.
2. Verify the expected output.

[Clean up]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Active Mode (POWER CONDITION = 01h)</td></tr>
<tr><td>Input parameter values</td><td>START STOP UNIT
LUN field in UPIU = D0h (Device well-known logical unit),
IMMED = 0b, POWER CONDITION MODIFIER = 0000b,
POWER CONDITION = 05h, NO_FLUSH = 0b, LOEJ = 0b,
START = 0b, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The START STOP UNIT command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN CDB</td></tr>
</table></td></tr>
</table>

8.7.2 Test Case ID: UFS_PowerMode_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 7.2</td></tr>
<tr><td>Test Purpose</td><td>To verify device returns GOOD STATUS for REQUEST SENSE command when it is UFS Sleep Power Mode</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Sleep Mode (POWER CONDITION = 02h)
2. Issue Query Request Read Attribute to verify bCurrentPowerMode is set to UFS-Sleep mode.

[Main]:
1. Issue REQUEST SENSE command.
2. Verify the expected output.

[Clean up]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Active Mode (POWER CONDITION = 01h)</td></tr>
<tr><td>Input parameter values</td><td>REQUEST SENSE
LUN field in UPIU = D0h (Device well-known logical unit),
DESC = 0, Allocation Length = 12h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The REQUEST SENSE command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
REQUEST SENSE command shall return pollable sense data with the sense key set to NOT READY (02h) and the additional sense code set to LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED</td></tr>
</table>

8.7.3 Test Case ID: UFS_PowerMode_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 7.2</td></tr>
<tr><td>Test Purpose</td><td>To verify device returns Success for Query Request Read Attribute command when it is in UFS Sleep Power Mode</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Sleep Mode (POWER CONDITION = 02h)

[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the Expected Output.

[Clean up]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Active Mode (POWER CONDITION = 01h)</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 02h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>02h</td><td>Success (00h)</td><td>22h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.7.4 Test Case ID: UFS_PowerMode_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 7.2</td></tr>
<tr><td>Test Purpose</td><td>To verify that device terminates START STOP UNIT command with CHECK CONDITION status if the POWER CONDITION field value is invalid while it is in UFS Power Down Mode</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Power Down Mode (POWER CONDITION = 03h)
2. Issue Query Request Read Attribute to verify bCurrentPowerMode is set to UFS-PowerDown mode.

[Main]:
1. Issue START STOP UNIT command.
2. Verify the expected output.

[Clean up]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Active Mode (POWER CONDITION = 01h)</td></tr>
<tr><td>Input parameter values</td><td>START STOP UNIT
LUN field in UPIU = D0h (Device well-known logical unit),
IMMED = 0b, POWER CONDITION MODIFIER = 0000b,
POWER CONDITION = 5h, NO_FLUSH = 0b, LOEJ = 0b,
START = 0b, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The START STOP UNIT command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ILLEGAL REQUEST</td><td>INVALID FIELD IN CDB</td></tr>
</table></td></tr>
</table>

8.7.5 Test Case ID: UFS_PowerMode_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 7.2</td></tr>
<tr><td>Test Purpose</td><td>To verify device returns GOOD STATUS for REQUEST SENSE command when it is in UFS Power Down Mode</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Power Down Mode (POWER CONDITION = 03h)
2. Issue Query Request Read Attribute to verify bCurrentPowerMode is set to UFS-PowerDown mode.

[Main]:
1. Issue REQUEST SENSE command.
2. Verify the expected output.

[Clean up]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Active Mode (POWER CONDITION = 01h)</td></tr>
<tr><td>Input parameter values</td><td>REQUEST SENSE
LUN field in UPIU = D0h (Device well-known logical unit),
DESC = 0, Allocation Length = 12h, CONTROL = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. The REQUEST SENSE command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
REQUEST SENSE command shall return pollable sense data with the sense key set to NOT READY (02h) and the additional sense code set to LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED</td></tr>
</table>

8.7.6 Test Case ID: UFS_PowerMode_06

<table>
<tr><td>Ref. specs Section</td><td>UFS Spec: 7.2</td></tr>
<tr><td>Test Purpose</td><td>To verify device returns Success for QUERY REQUEST READ ATTRIBUTE command when it is in UFS Power Down Mode</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Power Down Mode (POWER CONDITION = 03h)

[Main]:
1. Issue Query Request Read Attribute command.
2. Verify the expected output.

[Clean up]:
1. Issue START STOP UNIT command to Device well known LU to make the device power mode as UFS Active Mode (POWER CONDITION = 01h)</td></tr>
<tr><td>Input parameter values</td><td>Query Request UPIU: Read Attribute
Query Function = 01h, OPCODE = 03h, ATTRIBUTE IDN = 02h, INDEX = 00h, SELECTOR = 00h.</td></tr>
<tr><td>Expected Output</td><td>1. Query response shall be:
<table>
<tr><th>OPCODE</th><th>ATTRIBUTE IDN</th><th>Query Response</th><th>ATTRIBUTE VALUE</th></tr>
<tr><td>READ ATTRIBUTE (03h)</td><td>02h</td><td>Success (00h)</td><td>33h</td></tr>
</table>
Query Function = 01h, INDEX = 00h, SELECTOR = 00h</td></tr>
</table>

8.8 UFS RPMB

8.8.1 Test Case Id: UFS_RPMB_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 12.4.3.5 and 12.5.2.6</td></tr>
<tr><td>Test Purpose</td><td>To verify status of authenticated read operation when authentication key is not programmed.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Authentication key is not programmed.

[Main]:
1. Issue the SECURITY PROTOCOL OUT command.<a>
2. Verify the expected output.<1>
3. Issue the SECURITY PROTOCOL IN command.
4. Verify the expected output.<2>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC/Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>0</td><td>0</td><td>N (See note below)</td><td>0</td><td>0</td><td>0001h</td><td>0</td><td>0004h (Authenticated data read request)</td></tr>
</table>
NOTE:
The Nonce (N) shall be set to 0A1B2C3D4E5F60718293A4B5C6D7E8F9 (N[0] = 0x0A, N[15] = 0xF9).

b) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512</td></tr>
</table>

Expected Output

2. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

3. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
Any value	Any value	N	0	0	0001h	0007h	0400h

8.8.2 Test Case Id: UFS_RPMB_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 12.4.3.5 and 12.5.2.5</td></tr>
<tr><td>Test Purpose</td><td>To verify status of authenticated write operation when authentication key is not programmed.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Authentication key is not programmed.

[Main]:
1. Issue the Security Protocol Out command.<a>
2. Verify the expected output.<1>
3. Issue the Security Protocol Out command.
4. Verify the expected output.<2>
5. Issue Security Protocol In command.<c>
6. Verify the expected output.<3>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC/Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>0</td><td>All bytes set to 0Ah</td><td>0</td><td>0</td><td>0</td><td>0001h</td><td>0</td><td>0003h (Authenticated data Write request)</td></tr>
</table>
b) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0005h (Result read request)</td></tr>
</table>
c) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512</td></tr>
</table>

Expected Output

1. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

2. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

3. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
Any value	0	0	0	0	0	0007h	0300h

8.8.3 Test Case Id: UFS_RPMB_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 12.4.3.5 and 12.5.2.3</td></tr>
<tr><td>Test Purpose</td><td>To verify status of authentication key programming.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Authentication key is not programmed.

[Main]:
1. Issue the Security Protocol Out command.<a>
2. Verify the expected output.<1>
3. Issue the Security Protocol Out command.
4. Verify the expected output.<2>
5. Issue Security Protocol In command.<c>
6. Verify the expected output.<3>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>K (See note below)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0001h (Authenticationk programming request)</td></tr>
</table>
NOTE:
The authentication key (K) may be set to 0123456789ABCDEF01….EF (K[0] = 0x01, K[31] = 0xEF).

b) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0005h (Result read request)</td></tr>
</table>
c) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 00, ALLOCATION LENGTH = 512</td></tr>
</table>

Expected Output

1. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

2. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

3. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
0	0	0	0	0	0	0	0100h

8.8.4 Test Case Id: UFS_RPMB_04

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 12.4.3.5 and 12.5.2.3</td></tr>
<tr><td>Test Purpose</td><td>To verify that Authentication Key is not modifiable.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. Authentication Key is already programmed. (This can be verified by issuing RPMB Read/ Write request with valid inputs and checking that device returns “Operation OK” in result field).
2. Write counter is not expired.

[Main]:
1. Issue the Security Protocol Out command.<a>
2. Verify the expected output.<1>
3. Issue the Security Protocol Out command.
4. Verify the expected output.<2>
5. Issue Security Protocol In command.<c>
6. Verify the expected output.<3>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>Any value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0001h (Authentication key programming request)</td></tr>
</table>
b) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0005h (Result read request)</td></tr>
</table>
c) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512</td></tr>
</table>

Expected Output

1. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

2. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

3. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
0	0	0	0	0	0	0001h	0100h

8.8.5 Test Case Id: UFS_RPMB_05

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 12.4.3.5 and 12.5.2.4</td></tr>
<tr><td>Test Purpose</td><td>To verify that device is returning the write counter value.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. The authentication key (K) shall be set to 0123456789ABCDEF01….EF (K[0] = 0x01, K[31] = 0xEF).
2. No authenticated write operation is performed since authentication key has been programmed.

[Main]:
1. Issue the Security Protocol Out command.<a>
2. Verify the expected output.<1>
3. Issue Security Protocol In Command.
4. Verify the expected output.<2>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>0</td><td>0</td><td>N (See note below)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0002h (Reading of the write counter value request)</td></tr>
</table>
NOTE:
The Nonce (N) shall be set to 0A1B2C3D4E5F60718293A4B5C6D7E8F9 (N[0] = 0x0A, N[15] = 0xF9).

b) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512</td></tr>
</table>

Expected Output

1. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

2. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
M (See note below)	0	N	0h	0	0	0	0200h

NOTE:

The MAC (M) shall be

F454D07B16A82FBBE843FCC8CBA064AA3BD79F7C357B7EDC1312A304A395A575

(M[0] = 0x75, M[31] = 0xF4)

8.8.6 Test Case Id: UFS_RPMB_06

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 12.4.3.5 and 12.5.2.4</td></tr>
<tr><td>Test Purpose</td><td>To verify that device is returning the write counter value as FFFFFFFFh when the counter is expired.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. The authentication key (K) shall be set to 0123456789ABCDEF01….EF (K[0] = 0x01, K[31] = 0xEF)
2. Write Counter is expired. (i.e. Authenticated write operation is performed for FFFFFFFFh times)

[Main]:
1. Issue the Security Protocol Out command.<a>
2. Verify the expected output.<1>
3. Issue Security Protocol In Command.
4. Verify the expected output.<2>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>0</td><td>0</td><td>N (See note below)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0002h (Reading of the write counter value request)</td></tr>
</table>
NOTE:
The Nonce (N) shall be set to 0A1B2C3D4E5F60718293A4B5C6D7E8F9 (N[0] = 0x0A, N[15] = 0xF9)

b) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512</td></tr>
</table>

Expected Output

1. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

2. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
M	0	N	FFFFFFFFh	0	0	0080h	0200h

NOTE:

The MAC (M) shall be

D41A958467FDBB08235498C05D9DFE0BBAB2F0B9C18A1C3471ECB9A1D6F30B86

(M[0] = 0x86, M[31] = 0xD4)

8.8.7 Test Case Id: UFS_RPMB_07

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 12.4.3.5 and 12.5.2.5</td></tr>
<tr><td>Test Purpose</td><td>To validate status of authenticated write operation when write counter is expired (FFFFFFFFh)</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. The authentication key (K) shall be set to 0123456789ABCDEF01….EF (K[0] = 0x01, K[31] = 0xEF).
2. Write Counter is expired.(i.e. Authenticated write operation is performed for FFFFFFFFh times)

[Main]:
1. Issue the Security Protocol Out command.<a>
2. Verify the expected output.<1>
3. Issue the Security Protocol Out command.
4. Verify the expected output.<2>
5. Issue the Security Protocol In command.<c>
6. Verify the expected output.<3>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>M (See note below)</td><td>D (See note below)</td><td>0</td><td>FFFFFFFFh</td><td>0001h</td><td>0001h</td><td>0</td><td>0003h (Authenticated data write request)</td></tr>
</table>
NOTE:
1) The MAC (M) shall be set to BC576FBB28570814422DD5E7642306250AAD63E11BDF207F88859B96564CD9D5 (M[0] = 0xD5, M[31] = 0xBC).
2) The Data (D) shall be set to 000102030405…..FEFF (D[0] = 0x00, D[255] = 0xFF).

b) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0005h (Result read request)</td></tr>
</table>
c) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512</td></tr>
</table>

Expected Output

1. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

2. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

3. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
M1 (See note below)	0	0	FFFFFFFh	0001h	0	0085h	0300h

NOTE:
1) The MAC (M1) shall be set to C2BA34ADF3E768094F6598D63AA997216BB672E9EB4960450DE013C2862D98FE (M1[0] = 0xFE, M1[31] = 0xC2)

8.8.8 Test Case Id: UFS_RPMB_08

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 12.4.3.5 and 12.5.2.5</td></tr>
<tr><td>Test Purpose</td><td>To verify status of authenticated Write operation when write counter is not expired (FFFFFFFFh)</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. The authentication key (K) shall be set to 0123456789ABCDEF01….EF (K[0] = 0x01, K[31] = 0xEF).
2. Write counter is not expired.

[Main]:
1. Issue the Security Protocol Out command.<a>
2. Verify the expected output.<1>
3. Issue the Security Protocol In command.
4. Verify the expected output.<2>
5. Issue the Security Protocol Out command.<c>
6. Verify the expected output.<3>
7. Issue Security Protocol out command.<d>
8. Verify the expected output.<4>
9. Issue the Security Protocol In command.<e>
10. Verify the expected output.<5>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>0</td><td>0</td><td>N (See note below)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0002h (Reading of the write counter value request)</td></tr>
</table>
NOTE The Nonce (N) shall be set to 0A1B2C3D4E5F60718293A4B5C6D7E8F9 (N[0] = 0x0A, N[15] = 0xF9)

b) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512

c) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>Valid MAC</td><td>All bytes set to 0Ah</td><td>0</td><td>W</td><td>0001h</td><td>0001h</td><td>0</td><td>0003h (Authenticated data write request)</td></tr>
</table>
NOTE The MAC shall be calculated as per section 12.4.4.2 of UFS specification.

d) SECURITY PROTOCOL OUT:</td></tr>
</table>

SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, Transfer Length = 512

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
0	0	0	0	0	0	0	0005h (Result read request)

e) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512

Expected Output

1. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

2. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
Valid MAC	0	N	W	0	0	0	0200h

3. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

4. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
Valid MAC	0	0	W+1	0001h	0	0h (If Write Counter is not FFFFh) or 0080h (If Write Counter is FFFFh)	0300h

NOTE The MAC shall be calculated as per section 12.4.4.2 of UFS specification.

8.8.9 Test Case Id: UFS_RPMB_09

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 12.4.3.5 and 12.5.2.5</td></tr>
<tr><td>Test Purpose</td><td>To validate status of authenticated write operation with invalid address.</td></tr>
<tr><td>Test Procedure</td><td>
[Pre-condition]:
1. The authentication key (K) shall be set to 0123456789ABCDEF01….EF (K[0] = 0x01, K[31] = 0xEF)
2. Write counter is not expired.

[Main]:
1. Issue the Security Protocol Out command.<a>
2. Verify the expected output.<1>
3. Issue the Security Protocol In command.
4. Verify the expected output.<2>
5. Issue the Security Protocol Out command.<c>
6. Verify the expected output.<3>
7. Issue Security protocol out command.<d>
8. Verify the expected output.<4>
9. Issue the Security Protocol In command.<e>
10. Verify the expected output.<5>

[Clean up]:
None
</td></tr>
<tr><td>Input parameter values</td><td>
a) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, Transfer Length = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/Response</th></tr>
<tr><td>0</td><td>0</td><td>N (See note below)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0002h (Reading of the write counter value request)</td></tr>
</table>
NOTE The Nonce (N) shall be set to 0A1B2C3D4E5F60718293A4B5C6D7E8F9 (N[0] = 0x0A, N[15] = 0xF9)

b) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 00, ALLOCATION LENGTH = 512

c) SECURITY PROTOCOL OUT:
</td></tr>
</table>

SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, Transfer Length = 512

RPMB Data Frame 1:

MAC/Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
0	All bytes set to 0Ah	0	W	FFFEh	0003h	0	0003h (Authenticated data write request)

RPMB Data Frame 2:

MAC/Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/Response
0	All bytes set to 0Ah	0	W	FFFEh	0003h	0	0003h (Authenticated data write request)

RPMB Data Frame 3:

MAC/Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/Response
Valid MAC	All bytes set to 0Ah	0	W	FFFEh	0003h	0	0003h (Authenticated data write request)

NOTE The MAC shall be calculated as per section 12.4.4.2 of UFS specification.

d) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, Transfer Length = 512

RPMB Data Frame:

MAC/Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/Response
0	0	0	0	0	0	0	0005h (Result read request)

e) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512

Expected Output

1. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

2. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
0	0	N	W	0	0	0h	0200h

3. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

4. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

Key	Nonce	Write counter	Address	Block Count	Result	Request/Response
Valid MAC	0	W	FFFEh	0	0004h	0300h

NOTE The MAC shall be calculated as per section 12.4.4.2 of UFS specification.

8.8.10 Test Case Id: UFS_RPMB_10

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 12.4.3.5 and 12.5.2.6</td></tr>
<tr><td>Test Purpose</td><td>To verify status of authenticated read operation.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
1. The authentication key (K) shall be set to 0123456789ABCDEF01….EF (K[0] = 0x01, K[31] = 0xEF).
2. No authenticated write operation is performed since authentication key has been programmed.

[Main]:
1. Issue the Security Protocol Out command.<a>
2. Verify the expected output.<1>
3. Issue Security Protocol Out command.
4. Verify the expected output.<2>
5. Issue the Security Protocol In command.<c>
6. Verify the expected output.<3>
7. Issue Security Protocol out command.<d>
8. Verify the expected output.<4>
9. Issue the Security Protocol In command.<e>
10. Verify the expected output.<5>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, TRANSFER LENGTH = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>M (See Note 1)</td><td>D (See Note 2)</td><td>0</td><td>0</td><td>0005h</td><td>0001h</td><td>0</td><td>0003h (Authenticated data write request)</td></tr>
</table>
NOTE 1 The MAC (M) shall be set to 4A71F9A754CC322118344907D6C4D21D581B2DBB3715372D70F2B0CE20DA4AC1 (M[0] = 0xC1, M[31] = 0x4A).
NOTE 2 The Data (D) shall be set to 000102030405…..FEFF (D[0] = 0x00, D[255] = 0xFF).

b) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, Transfer Length = 512

RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0005h (Result read request)</td></tr>
</table>
c) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512</td></tr>
</table>

d) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, Transfer Length = 512

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
0	0	N (See note below)	0	0005h	0001h	0	0004h (Authenticated data read request)

NOTE The Nonce (N) shall be set to 0A1B2C3D4E5F60718293A4B5C6D7E8F9 (N[0] = 0x0A, N[15] = 0xF9)

e) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512

Expected Output

1. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

2. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

3. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response
M1 (See Note)	0	0	00000001h	0005h	0	0	0300h

NOTE The MAC (M1) shall be F3D835867CC8123E729598B162A27231AB2B5A59CCB6C199B670167C0C9A9D7F (M1[0] = 0x7F, M1[31] = 0xF3)

4. The SECURITY PROTOCOL OUT command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. The SECURITY PROTOCOL IN command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

RPMB Data Frame:

MAC /Key	Data	Nonce	Write counter	Address	Block Count	Result	Request/ Response

	M2 (See Note)	D	N	0	0005h	0001h	0	0400h

NOTE The MAC (M2) shall be 9C8CD67F7943BCB241471A53F8C30241DF0BD18A125D9255B31A512FDE491C6A (M2[0] = 0x6A, M2[31] = 0x9C).

8.8.11 Test Case Id: UFS_RPMB_11

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 12.4.3.5 and 12.5.2.6</td></tr>
<tr><td>Test Purpose</td><td>To verify status of authenticated read operation with invalid address.</td></tr>
<tr><td>Test Procedure</td><td>[Pre-condition]:
1. The authentication key (K) shall be set to 0123456789ABCDEF01….EF (K[0] = 0x01, K[31] = 0xEF)

[Main]:
1. Issue Security protocol out command.<a>
2. Verify the expected output.<1>
3. Issue the Security Protocol In command.
4. Verify the expected output.<2>

[Clean up]:
None</td></tr>
<tr><td>Input parameter values</td><td>a) SECURITY PROTOCOL OUT:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, Transfer Length = 512

RPMB Data Frame :
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>FFFEh</td><td>3</td><td>0</td><td>0004h (Authenticated data read request)</td></tr>
</table>
b) SECURITY PROTOCOL IN:
SECURITY PROTOCOL = ECh, SECURITY PROTOCOL SPECIFIC = 0001h, INC_512 = 0, ALLOCATION LENGTH = 512</td></tr>
<tr><td>Expected Output</td><td>1. The SECURITY PROTOCOL OUT command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The SECURITY PROTOCOL IN command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
RPMB Data Frame:
<table>
<tr><th>MAC /Key</th><th>Data</th><th>Nonce</th><th>Write counter</th><th>Address</th><th>Block Count</th><th>Result</th><th>Request/ Response</th></tr>
<tr><td>Valid MAC</td><td>Any value</td><td>0</td><td>0</td><td>FFFEh</td><td>0003h</td><td>0004h (If Write Counter is not FFFFFFFFh)or 0084h (If Write Counter is FFFFFFFFh)</td><td>0400h</td></tr>
</table>
NOTE The MAC shall be calculated as per section 12.4.4.2 of UFS specification.</td></tr>
</table>

8.9 UFS UPIU

8.9.1 Test Case Id: UFS_UPIU_01

Ref. specs Section	UFS: Section 14.3
Test Purpose	To verify that device returns NOP IN when sending NOP OUT command from host
Test Procedure	**[Precondition]:** None. **[Main]:** 1. Issue NOP OUT UPIU 2. Verify the Expected Output.. **[Clean up]:** None
Input parameter values	**NOP OUT UPIU** Transaction Type = 00000000b, Flags = 00h, Data Segment Length = 0000h
Expected Output	1. The Device shall return NOP IN UPIU as following: Transaction Type = 00100000b, Flag = 00h, Response = 00h, Device Information = 00h, Data Segment Length = 0000h

8.9.2 Test Case Id: UFS_UPIU_02

Ref. specs Section	UFS: Section 14.3
Test Purpose	To verify that device returns REJECT UPIU when it receives an UPIU with invalid Transaction Type
Test Procedure	**[Precondition]:** None. **[Main]:** 1. Issue invalid UPIU 2. Verify the Expected Output.. **[Clean up]:** None
Input parameter values	**Invalid UPIU** Transaction Type = 00011111b, Flags = 00h, Data Segment Length = 0000h
Expected Output	1. The Device shall return REJECT UPIU as following: Transaction Type = 00111111b, Flag = 00h, Response = 01h, Device Information = 00h, Data Segment Length = 0000h, Basic Header Status = 01h

8.10 UFS UPIU Flags

8.10.1 Test Case Id: UFS_UPIU_Flags_01

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.3.2</td></tr>
<tr><td>Test Purpose</td><td>To verify Flag.W and Flag.R when WRITE (6) and READ (6) are issued.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue WRITE (6) Command.
2. Verify expected output<1>.
3. Issue READ (6) Command.
4. Verify expected output<2>.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>WRITE (6):
LBA = 0h, TRANSFER LENGTH = 1h, CONTROL = 00h,
Flags.W = 1 and Flags.R = 0

READ (6):
LBA = 0h, TRANSFER LENGTH = 1h, CONTROL = 00h,
Flags.W = 0 and Flags.R = 1</td></tr>
<tr><td>Expected Output</td><td>1. The Write (6) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
2. The Read (6) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
3. Read data from LBA 0 shall be the same data as write data to LBA 0.</td></tr>
</table>

8.10.2 Test Case Id: UFS_UPIU_Flags_02

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.4.2 and 13.4.12</td></tr>
<tr><td>Test Purpose</td><td>To verify that the device sets Flag.D in Response UPIU.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue WRITE (10) command.
2. Send Data Out UPIU.
3. Verify expected output.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>1) WRITE (10):
LBA = 00h, TRANSFER LENGTH = 01h, WRPROTECT = 0b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h

2) Data Out UPIU:
Data Transfer Count and Data Segment Length shall be set to Data Transfer Count value in the received RTT UPIU minus 04h.</td></tr>
<tr><td>Expected Output</td><td>1. The WRITE (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ABORTED COMMAND</td><td>-</td></tr>
</table>
2. Flags.D in Response UPIU shall be 1b.</td></tr>
</table>

8.10.3 Test Case Id: UFS_UPIU_Flags_03

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 10.5.4.2 and 13.4.12</td></tr>
<tr><td>Test Purpose</td><td>To verify that the device sets Flag.D in Response UPIU.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue WRITE (10) Command.
2. Send Data Out UPIU.
3. Verify expected output.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>1) WRITE (10):
LBA = 0h, TRANSFER LENGTH = 01h,, WRPROTECT = 0b, DPO = 0b, FUA = 0b, FUA_NV = 0b, GROUP NUMBER = 00h, CONTROL = 00h.

2) Data Out UPIU:
Data Transfer Count and Data Segment Length shall be set to Data Transfer Count value in the received RTT UPIU plus 04h.</td></tr>
<tr><td>Expected Output</td><td>1. The Write (10) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>ABORTED COMMAND</td><td>-</td></tr>
</table>
2. Flags.D in Response UPIU shall be 1b.</td></tr>
</table>

8.11 UFS Unit Attention Condition

8.11.1 Test Case Id: UFS_Unit_Attention_01

Ref. specs Section	UFS: Section 11.3.12 [SAM] : Section 5.14
Test Purpose	To verify not clearing Unit Attention Condition by REPORT LUNS command after power on reset
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. Power (VCC, VCCQ and VCCQ2) down. 2. Power (VCC, VCCQ and VCCQ2) up. 3. Issue NOP OUT UPIU until receiving NOP IN UPIU. 4. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 5. Verify expected output<1>. 6. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 7. Verify expected output<2> or <3>. If Output is <2>, then back to 6. If Output is <3>, then go to 8. 8. Issue REPORT LUNS command. 9. Verify expected output<4>. 10. Issue TEST UNIT READY command. 11. Verify expected output<5>. 12. Repeat steps 10 and 11 for all configured logical units. **[Clean up]** None
Input parameter values	**REPORT LUNS:** SELECT REPORT = 00h, ALLOCATION LENGTH = (number of LUNs*8)+ 8, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The REPORT LUNS command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or POWER ON RESET OCCURRED or NO ADDITIONAL SENSE CODE

8.11.2 Test Case Id: UFS_Unit_Attention_02

Ref. specs Section	UFS: Section 11.3.17 [SAM] : Section 5.14
Test Purpose	To verify clearing Unit Attention Condition by REQUEST SENSE command after power on reset.
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. Power (VCC, VCCQ and VCCQ2) down. 2. Power (VCC, VCCQ and VCCQ2) up. 3. Issue NOP OUT UPIU until receiving NOP IN UPIU. 4. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 5. Verify expected output<1>. 6. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 7. Verify expected output<2> or <3>. If Output is <2>, then back to 6. If Output is <3>, then go to 8. 8. Issue REQUEST SENSE command. 9. Verify expected output<4> in RESPONSE UPIU and output<5> in DATA IN UPIU. 10. Issue TEST UNIT READY command. 11. Verify expected output<6>. 12. Repeat steps 8 through 11 for all configured logical units. **[Clean up]** None
Input parameter values	**REQUEST SENSE:** DESC = 0, ALLOCATION LENGTH = 12h, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The REQUEST SENSE command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. The REQUES SENSE command shall provide 18 bytes sense data which is the following:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or POWER ON RESET OCCURRED or NO ADDITIONAL SENSE CODE

6. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

8.11.3 Test Case Id: UFS_Unit_Attention_03

Ref. specs Section	UFS: Section 11.3.5 [SAM] : Section 5.14
Test Purpose	To verify clearing Unit Attention Condition by READ (6) command after power on reset
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. Power (VCC, VCCQ and VCCQ2) down. 2. Power (VCC, VCCQ and VCCQ2) up. 3. Issue NOP OUT UPIU until receiving NOP IN UPIU. 4. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 5. Verify expected output<1>. 6. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 7. Verify expected output<2> or <3>. If Output is <2>, back to 6. If Output is <3>, go to 8. 8. Issue READ (6) command to a logical unit. 9. Verify expected output<4>. 10. Issue TEST UNIT READY command. 11. Verify expected output<5>. 12. Repeat steps 8 through 11 for all configured logical units. **[Clean up]** None
Input parameter values	**READ (6):** LBA = 0h, TRANSFER LENGTH = 1h, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The READ (6) command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or POWER ON RESET OCCURRED or NO ADDITIONAL SENSE CODE

5. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

8.11.4 Test Case Id: UFS_Unit_Attention_04

Ref. specs Section	UFS: Section 11.3.12 [SAM] : Section 5.14
Test Purpose	To verify not clearing Unit Attention Condition by REPORT LUNS command after hardware reset
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. Hardware reset. 2. Issue NOP OUT UPIU until receiving NOP IN UPIU. 3. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 4. Verify expected output<1>. 5. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 6. Verify expected output<2> or <3>. If Output is <2>, then back to 5. If Output is <3>, then go to 7. 7. Issue REPORT LUNS command. 8. Verify expected output<4>. 9. Issue TEST UNIT READY command. 10. Verify expected output<5>. 11. Repeat steps 9 and 10 for all configured logical units. **[Clean up]** None
Input parameter values	**REPORT LUNS:** SELECT REPORT = 00h, ALLOCATION LENGTH = (number of LUNs*8)+ 8, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The REPORT LUNS command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or SCSI BUS RESET OCCURRED or NO ADDITIONAL SENSE CODE

8.11.5 Test Case Id: UFS_Unit_Attention_05

Ref. specs Section	UFS: Section 11.3.17 [SAM] : Section 5.14
Test Purpose	To verify clearing Unit Attention Condition by REQUEST SENSE command after hardware reset
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. Hardware reset. 2. Issue NOP OUT UPIU until receiving NOP IN UPIU. 3. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 4. Verify expected output<1>. 5. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 6. Verify expected output<2> or <3>. If Output is <2>, then back to 5. If Output is <3>, then go to 7. 7. Issue REQUEST SENSE command. 8. Verify expected output<4> in RESPONSE UPIU and output<5> in DATA IN UPIU. 9. Issue TEST UNIT READY command. 10. Verify expected output<6>. 11. Repeat steps 7 through 10 for all configured logical units. **[Clean up]** None
Input parameter values	**REQUEST SENSE:** DESC = 0, ALLOCATION LENGTH = 12h, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The REQUEST SENSE command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. The REQUES SENSE command shall provide 18 bytes sense data which is the following:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or SCSI BUS RESET OCCURRED or NO ADDITIONAL SENSE CODE

6. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

8.11.6 Test Case Id: UFS_Unit_Attention_06

Ref. specs Section	UFS: Section 11.3.5 [SAM] : Section 5.14
Test Purpose	To verify clearing Unit Attention Condition by READ (6) command after hardware reset
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. Hardware reset. 2. Issue NOP OUT UPIU until receiving NOP IN UPIU. 3. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 4. Verify expected output<1>. 5. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 6. Verify expected output<2> or <3>. If Output is <2>, then back to 5. If Output is <3>, then go to 7. 7. Issue READ (6) command to a logical unit. 8. Verify expected output<4>. 9. Issue TEST UNIT READY command. 10. Verify expected output<5>. 11. Repeat steps 7 through 10 for all configured logical units. **[Clean up]** None
Input parameter values	**READ (6):** LBA = 0h, TRANSFER LENGTH = 1h, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The READ (6) command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or SCSI BUS RESET OCCURRED or NO ADDITIONAL SENSE CODE

5. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

8.11.7 Test Case Id: UFS_Unit_Attention_07

Ref. specs Section	UFS: Section 11.3.12 [SAM] : Section 5.14
Test Purpose	To verify not clearing Unit Attention Condition by REPORT LUNS command after EndPointReset
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. EndPointReset. 2. Issue NOP OUT UPIU until receiving NOP IN UPIU. 3. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 4. Verify expected output<1>. 5. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 6. Verify expected output<2> or <3>. If Output is <2>, then back to 5. If Output is <3>, then go to 7. 7. Issue REPORT LUNS command. 8. Verify expected output<4>. 9. Issue TEST UNIT READY command. 10. Verify expected output<5>. 11. Repeat steps 9 and 10 for all configured logical units. **[Clean up]** None
Input parameter values	**REPORT LUNS:** SELECT REPORT = 00h, ALLOCATION LENGTH = (number of LUNs *8)+ 8, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The REPORT LUNS command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or SCSI BUS RESET OCCURRED or NO ADDITIONAL SENSE CODE

8.11.8 Test Case Id: UFS_Unit_Attention_08

Ref. specs Section	UFS: Section 11.3.17 [SAM] : Section 5.14
Test Purpose	To verify clearing Unit Attention Condition by REQUEST SENSE command after EndPointReset
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. EndPointReset. 2. Issue NOP OUT UPIU until receiving NOP IN UPIU. 3. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 4. Verify expected output<1>. 5. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 6. Verify expected output<2> or <3>. If Output is <2>, then back to 5. If Output is <3>, then go to 7. 7. Issue REQUEST SENSE command. 8. Verify expected output<4> in RESPONSE UPIU and output<5> in DATA IN UPIU. 9. Issue TEST UNIT READY command. 10. Verify expected output<6>. 11. Repeat steps 7 through 10 for all configured logical units. **[Clean up]** None
Input parameter values	**REQUEST SENSE:** DESC = 0, Allocation Length = 12h, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The REQUEST SENSE command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. The REQUES SENSE command shall provide 18 bytes sense data which is the following:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or SCSI BUS RESET OCCURRED or NO ADDITIONAL SENSE CODE

6. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

8.11.9 Test Case Id: UFS_Unit_Attention_09

Ref. specs Section	UFS: Section 11.3.5 [SAM] : Section 5.14
Test Purpose	To verify clearing Unit Attention Condition by READ (6) command after EndPointReset.
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. EndPointReset. 2. Issue NOP OUT UPIU until receiving NOP IN UPIU. 3. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 4. Verify expected output<1>. 5. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 6. Verify expected output<2> or <3>. If Output is <2>, then back to 5. If Output is <3>, then go to 7. 7. Issue READ (6) command to a logical unit. 8. Verify expected output<4>. 9. Issue TEST UNIT READY command. 10. Verify expected output<5>. 11. Repeat steps 7 through 10 for all configured logical units. **[Clean up]** None
Input parameter values	**READ (6):** LBA = 0h, TRANSFER LENGTH = 1h, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The READ (6) command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or SCSI BUS RESET OCCURRED or NO ADDITIONAL SENSE CODE

5. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

8.11.10 Test Case Id: UFS_Unit_Attention_10

Ref. specs Section	UFS: Section 7.1.5, 11.3.12 [SAM] : Section 5.14
Test Purpose	To verify not clearing Unit Attention Condition by REPORT LUNS command after reset a Host UniPro stack by DME_RESET.req and initiating Link Startup Sequence.
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. Host resets a Host UniPro stack by DME_RESET.req and initiates Link Startup Sequence. 2. Issue NOP OUT UPIU until receiving NOP IN UPIU. 3. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 4. Verify expected output<1>. 5. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 6. Verify expected output<2> or <3>. If Output is <2>, then back to 5. If Output is <3>, then go to 7. 7. Issue REPORT LUNS command. 8. Verify expected output<4>. 9. Issue TEST UNIT READY command. 10. Verify expected output<5>. 11. Repeat steps 9 and 10 for all configured logical units. **[Clean up]** None
Input parameter values	**REPORT LUNS:** SELECT REPORT = 00h, ALLOCATION LENGTH = (number of LUNs *8)+ 8, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The REPORT LUNS command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or I_T NEXUS LOSS OCCURRED or NO ADDITIONAL SENSE CODE

8.11.11 Test Case Id: UFS_Unit_Attention_11

Ref. specs Section	UFS: Section 7.1.5, 11.3.17 [SAM] : Section 5.14
Test Purpose	To verify clearing Unit Attention Condition by REQUEST SENSE command after reset a Host UniPro stack by DME_RESET.req and initiating Link Startup Sequence.
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. Host resets a Host UniPro stack by DME_RESET.req and initiates Link Startup Sequence. 2. Issue NOP OUT UPIU until receiving NOP IN UPIU. 3. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 4. Verify expected output<1>. 5. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 6. Verify expected output<2> or <3>. If Output is <2>, then back to 5. If Output is <3>, then go to 7. 7. Issue REQUEST SENSE command. 8. Verify expected output<4> in RESPONSE UPIU and output<5> in DATA IN UPIU. 9. Issue TEST UNIT READY command. 10. Verify expected output<6>. 11. Repeat steps 7 through 10 for all configured logical units. **[Clean up]** None
Input parameter values	**REQUEST SENSE:** DESC = 0, Allocation Length = 12h, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The REQUEST SENSE command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

5. The REQUES SENSE command shall provide 18 bytes sense data which is the following:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or I_T NEXUS LOSS OCCURRED or NO ADDITIONAL SENSE CODE

6. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

8.11.12 Test Case Id: UFS_Unit_Attention_12

Ref. specs Section	UFS: Section 7.1.5, 11.3.5 [SAM] : Section 5.14
Test Purpose	To verify clearing Unit Attention Condition by READ (6) command after reset a Host UniPro stack by DME_RESET.req and initiating Link Startup Sequence.
Test Procedure	**[Precondition]:** 1. bInitPowerMode in DEVICE DESCRIPTOR is set to 01h. **[Main]:** 1. Host resets a Host UniPro stack by DME_RESET.req and initiates Link Startup Sequence. 2. Issue NOP OUT UPIU until receiving NOP IN UPIU. 3. Issue QUERY REQUEST UPIU (SET FLAG fDeviceInit). 4. Verify expected output<1>. 5. Issue QUERY REQUEST UPIU (READ FLAG fDeviceInit). 6. Verify expected output<2> or <3>. If Output is <2>, then back to 5. If Output is <3>, then go to 7. 7. Issue READ (6) command to a logical unit. 8. Verify expected output<4>. 9. Issue TEST UNIT READY command. 10. Verify expected output<5>. 11. Repeat steps 7 through 10 for all configured logical units. **[Clean up]** None
Input parameter values	**READ (6):** LBA = 0h, TRANSFER LENGTH = 1h, CONTROL = 00h **TEST UNIT READY:** CONTROL = 00h

Expected Output

1. The QUERY RESPONSE UPIU for SET FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

2. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	01h

3. The QUERY RESPONSE UPIU for READ FLAG fDeviceInit shall be:

Query Response	FLAGS VALUE
Success	00h

4. The READ (6) command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Failure	CHECK CONDITION	UNIT ATTENTION	POWER ON, RESET, or BUS DEVICE RESET OCCURRED or I_T NEXUS LOSS OCCURRED or NO ADDITIONAL SENSE CODE

5. The TEST UNIT READY command response shall be:

Response	Status	Sense Key	Additional Sense Code
Target Success	GOOD	-	-

8.11.13 Test Case Id: UFS_Unit_Attention_13

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.12
[SAM] : Section 5.14</td></tr>
<tr><td>Test Purpose</td><td>To verify not clearing Unit Attention Condition by REPORT LUNS command after Logical unit reset.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Task Management Request UPIU for Logical Unit Reset.
2. Verify expected output<1>.
3. Issue REPORT LUNS command.
4. Verify expected output<2>.
5. Issue TEST UNIT READY command.
6. Verify expected output<3>.
7. Repeat steps 1 through 6 for all configured logical units.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>REPORT LUNS:
SELECT REPORT = 00h, ALLOCATION LENGTH = (number of LUNs *8)+ 8, CONTROL = 00h

TEST UNIT READY:
CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The TASK MANAGEMENT RESPONSE UPIU shall be:
<table>
<tr><th>Response</th><th>Task Management Service Response</th></tr>
<tr><td>Target Success</td><td>Task Management Function Complete (00h)</td></tr>
</table>
2. The REPORT LUNS command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
3. The TEST UNIT READY command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>UNIT ATTENTION</td><td>POWER ON, RESET, or
BUS DEVICE RESET OCCURRED or BUS DEVICE RESET FUNCTION OCCURRED or NO ADDITIONAL SENSE CODE</td></tr>
</table></td></tr>
</table>

8.11.14 Test Case Id: UFS_Unit_Attention_14

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.17
[SAM] : Section 5.14</td></tr>
<tr><td>Test Purpose</td><td>To verify clearing Unit Attention Condition by REQUEST SENSE command after Logical Unit reset.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Task Management Request UPIU for Logical Unit Reset.
2. Verify expected output<1>.
3. Issue REQUEST SENSE command.
4. Verify expected output<2> in RESPONSE UPIU and output<3> in DATA IN UPIU.
5. Issue TEST UNIT READY command.
6. Verify expected output<4>.
7. Repeat steps 1 through 6 for all configured logical units.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>REQUEST SENSE:
DESC = 0, Allocation Length = 12h, CONTROL = 00h

TEST UNIT READY:
CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The TASK MANAGEMENT RESPONSE UPIU shall be:
<table>
<tr><th>Response</th><th>Task Management Service Response</th></tr>
<tr><td>Target Success</td><td>Task Management Function Complete (00h)</td></tr>
</table>
2. The REQUEST SENSE command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table>
3. The REQUES SENSE command shall provide 18 bytes sense data which is the following:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>UNIT ATTENTION</td><td>POWER ON, RESET, or BUS DEVICE RESET OCCURRED or BUS DEVICE RESET FUNCTION OCCURRED or NO ADDITIONAL SENSE CODE</td></tr>
</table>
4. The TEST UNIT READY command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

8.11.15 Test Case Id: UFS_Unit_Attention_15

<table>
<tr><td>Ref. specs Section</td><td>UFS: Section 11.3.5
[SAM] : Section 5.14</td></tr>
<tr><td>Test Purpose</td><td>To verify clearing Unit Attention Condition by READ (6) command after Logical Unit reset.</td></tr>
<tr><td>Test Procedure</td><td>[Precondition]:
None

[Main]:
1. Issue Task Management Request UPIU for Logical Unit Reset.
2. Verify expected output<1>.
3. Issue READ (6) command to a logical unit.
4. Verify expected output<2>.
5. Issue TEST UNIT READY command.
6. Verify expected output<3>.
7. Repeat steps 1 through 6 for all configured logical units.

[Clean up]
None</td></tr>
<tr><td>Input parameter values</td><td>READ (6):
LBA = 0h, TRANSFER LENGTH = 1h, CONTROL = 00h

TEST UNIT READY:
CONTROL = 00h</td></tr>
<tr><td>Expected Output</td><td>1. The TASK MANAGEMENT RESPONSE UPIU shall be:
<table>
<tr><th>Response</th><th>Task Management Service Response</th></tr>
<tr><td>Target Success</td><td>Task Management Function Complete (00h)</td></tr>
</table>
2. The READ (6) command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Failure</td><td>CHECK CONDITION</td><td>UNIT ATTENTION</td><td>POWER ON, RESET, or BUS DEVICE RESET OCCURRED or DEVICE RESET FUNCTION OCCURRED or NO ADDITIONAL SENSE CODE</td></tr>
</table>
3. The TEST UNIT READY command response shall be:
<table>
<tr><th>Response</th><th>Status</th><th>Sense Key</th><th>Additional Sense Code</th></tr>
<tr><td>Target Success</td><td>GOOD</td><td>-</td><td>-</td></tr>
</table></td></tr>
</table>

Annex A (informative) Differences between JESD224A and JESD224

These tables briefly describe most of the differences between the text of this standard, JESD224A, and its predecessor JESD224 (March 2013).

Clause	Description of Change
2	Updated revisions and added JESD220-2.
7.3.3	Expected output, item 2, changed from : TST=111b, SWP=1b and BUSY TIMEOUT PERIOD = FFFF, to TST=000b, SWP=1b and 0000h.
7.3.7	Expected output, removed item 2 (Only the value of the following fields shall be verified: READ RETRY COUNT= FFh, WRITE RETRY COUNT = FFh, RECOVERY TIME LIMIT = FFFFh).
7.3.14	Expected output, added note.
7.4.17	Expected output, under Additional Sense Code, changed CDB to Parameter list.
7.8.2	Content removed.
7.16.9	Expected output, item 2, changed 18h to 20h.
8.3.1	Input parameter values, item b) changed: Length = 1Fh to Length = 40h.
8.4.1	Input parameter values, changed: Length = 1Fh to Length = 40h.
8.5.19	Expected output, item 2, under Flag Value, added “or 0b”.
8.5.21	Expected output, item 2, under Query Response, added “or Invalid Value (FAh)”.
8.5.23	Expected output, item 2, under Flag Value, added “0b or”.
8.6.19	Expected output, item 2, under Attribute Value, replaced “N” with “FFh”.
8.6.21	Expected output, item 2, under Attribute Value, replaced “N” with “01h”.
8.6.23	Expected output, item 2, under Attribute Value, replaced “N” with “10h”.
8.6.24	Expected output, item 2, under Attribute Value, replaced “01h” with “0h”.
8.6.25	Expected output, item 1, under Attribute Value, replaced “N” with “FFh”.
8.6.26	Expected output, item 1, under Attribute Value, removed “00h, 02h, or 03h”.
8.6.27	Expected output, item 1, under Attribute Value, replaced “00h – 05h” with “01h”.
8.6.30	Expected output, item 1, under Attribute Value, replaced “N” with “bMaxInBufferSize + 01h”.
8.6.33	Expected output, item 1, under Attribute Value, replaced “N” with “bMaxInBufferSize + 01h”.
8.6.34	Expected output, item 1, under Attribute Value, replaced “Any Value” with “01h”.
8.6.35	Content Removed.
8.6.36	Test Purpose, changed “write once” to “persistent”.
8.6.36	Expected output, item 1, under Attribute Value, replaced “N” with “FCh”.
8.6.38	Expected output, item 1, under Attribute Value, replaced “00” with “FFh”.
8.6.40	Expected output, item 1, under Attribute Value, replaced “0000h – 0007h” with “FFFFh”.
8.6.41	Expected output, item 1, under Attribute Value, replaced “Any Value” with “FFFFh”.
8.6.42	Test Procedure, under Precondition, removed “none” added content.
8.6.43	Removed “Test Case Id: UFS_QR_WriteAttribute_26”. Renumbered 8.6.44 to 8.6.43.
8.6.43	Expected output, item 1, under Attribute Value, replaced “Any Value” with “01h”.

Standard Improvement Form JEDEC JESD224A

The purpose of this form is to provide the Technical Committees of JEDEC with input from the industry regarding usage of the subject standard. Individuals or companies are invited to submit comments to JEDEC. All comments will be collected and dispersed to the appropriate committee(s).

If you can provide input, please complete this form and return to:

JEDEC
Attn: Publications Department
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

Fax: 703.907.7583

1. I recommend changes to the following:

☐ Requirement, clause number _____

☐ Test method number _____ Clause number _____

The referenced clause number has proven to be:

☐ Unclear ☐ Too Rigid ☐ In Error

☐ Other _____

2. Recommendations for correction:

_____

_____

_____

_____

3. Other suggestions for document improvement:

_____

_____

_____

_____

Submitted by

Name: _____ Phone: _____

Company: _____ E-mail: _____

Address: _____

City/State/Zip: _____ Date: _____


JEDEC