Power Delivery Compliance Plan
for the Power Delivery Specification Revision 2.0
Version 1.2

Revision 1.02, Version 2.0
8 March 2017

Revision History

Revision	Version	Issue Date	Comment
1.0	2.0	June 9, 2016	Initial release Revision 1.0
1.01	2.0	October 26, 2016	Removed " PROT_SR_HR" from Section 7 and Table 12.2
1.02	2.0	March 8, 2016	Added Eye Diagram section 5.5

Copyright © 2011 - 2016, USB Implementers Forum, Inc.

All rights reserved.

A LICENSE IS HEREBY GRANTED TO REPRODUCE THIS SPECIFICATION FOR INTERNAL USE ONLY. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, IS GRANTED OR INTENDED HEREBY.

USB-IF AND THE AUTHORS OF THIS SPECIFICATION EXPRESSLY DISCLAIM ALL LIABILITY FOR INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, RELATING TO IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. USB-IF AND THE AUTHORS OF THIS SPECIFICATION ALSO DO NOT WARRANT OR REPRESENT THAT SUCH IMPLEMENTATION(S) WILL NOT INFRINGE THE INTELLECTUAL PROPERTY RIGHTS OF OTHERS.

THIS SPECIFICATION IS PROVIDED "AS IS" AND WITH NO WARRANTIES, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE. ALL WARRANTIES ARE EXPRESSLY DISCLAIMED. NO WARRANTY OF MERCHANTABILITY, NO WARRANTY OF NON-INFRINGEMENT, NO WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE, AND NO WARRANTY ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.

IN NO EVENT WILL USB-IF OR USB-IF MEMBERS BE LIABLE TO ANOTHER FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA OR ANY INCIDENTAL, CONSEQUENTIAL, INDIRECT, OR SPECIAL DAMAGES, WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THE USE OF THIS SPECIFICATION, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

All product names are trademarks, registered trademarks, or servicemarks of their respective owners.

Contributors

Name	Company
James Choate	Agilent Technologies, Inc.
Bill Cornelius	Apple
Hsiao-Ping Jennifer Tsai	Apple
Anup Nayak	Cypress Semiconductor
Andrea Colognese	Canova Tech
Matteo Casalin	Canova Tech
Richard Petrie	DisplayLink (UK) Ltd.
Pete Burges	DisplayLink (UK) Ltd.
Chuck Trefts	Ellisys
Chien-Cheng Kuo	Etron Technology, Inc.
Gwo Hwa Wang	Fairchild Semiconductor
Oscar Freitas	Fairchild Semiconductor
Miller Lin	Genesys Logic, Inc.
Mike Engbretson	Granite River Labs
Tim McKee	Intel Corporation
Rahman Ismail	Intel Corporation
Brad Saunders	Intel Corporation
Bob Dunstan	Intel Corporation
Mike Micheletti	LeCroy Corporation
Mick Davis	Microchip
William Chiechi	Microchip
Shannon Cash	Microchip
Josh Averyt	Microchip
Ben Crowe	MQP Electronics Ltd.
Pat Crowe	MQP Electronics Ltd.
Sten Carlsen	MQP Electronics Ltd.
Krishnan TN	NXP Semiconductors

Vijendra Kuroodi	NXP Semiconductors
Kiichi Muto	Renesas Electronics Corp.
Hajime Nozaki	Renesas Electronics Corp.
Philip Leung	Renesas Electronics Corp.
Takashi Sato	ROHM Co., Ltd.
Akihiro Ono	ROHM Co., Ltd.
Yoshinori Ohwaki	ROHM Co., Ltd.
Chris Lin	ROHM Co., Ltd.
Vijendra Kuroodi	ROHM Co., Ltd.
Alvin Cox	Seagate Technology LLC
Diane Rose	Specwerkz
Nicolas Florenchie	ST-Ericsson
Pascal Legrand	STMicroelectronics
Christophe Lorin	STMicroelectronics
Nathalie Ballot	STMicroelectronics
Gajendra Patro	Tektronix, Inc.
Srikrishna N.H.	Tektronix, Inc.
Deric Waters	Texas Instruments
Lee Myers	Texas Instruments
Bill Waters	Texas Instruments
Fong-Jim Wang	VIA Technologies, Inc.
Terrance Shih	VIA Technologies, Inc.

1 INTRODUCTION

1.1 Power Delivery

USB has evolved from a data interface capable of supplying limited power to a primary provider of power with a data interface. Today many devices charge or get their power from USB ports contained in laptops, cars, aircraft or even wall sockets. USB has become a ubiquitous power socket for many small devices such as cell phones, MP3 players and other hand-held devices. Users need USB to fulfil their requirements not only in terms of data but also to provide power to, or charge, their devices simply, often without the need to load a driver, in order to carry out "traditional" USB functions.

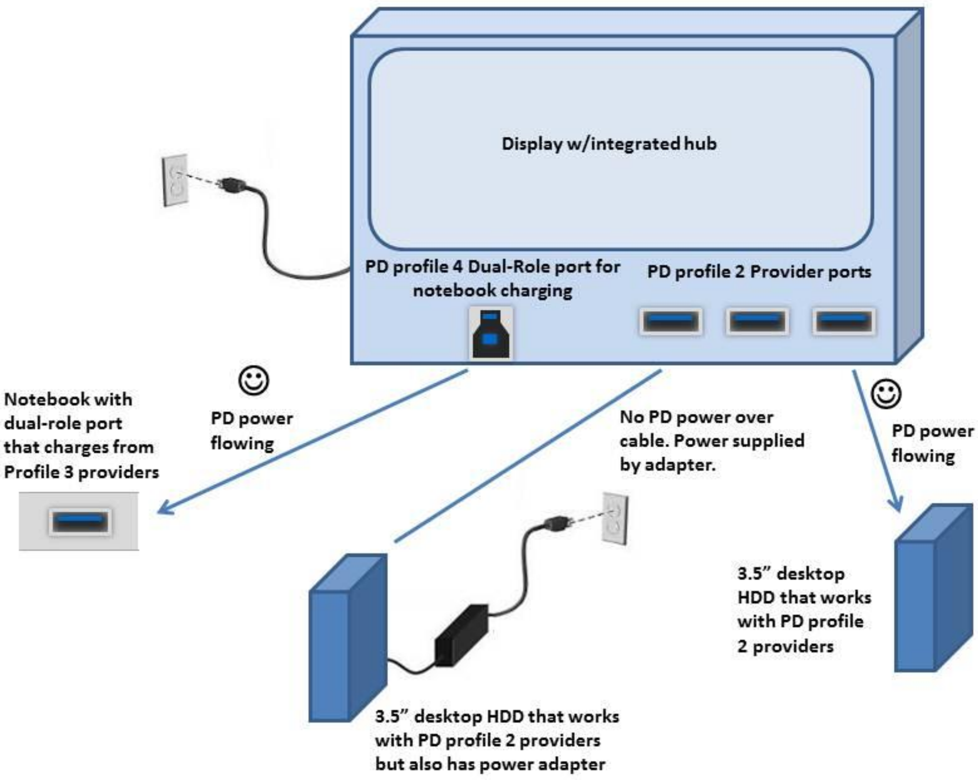
The USB Power Delivery Specification enables the maximum functionality of USB by providing more flexible power delivery along with data over a single cable. Its aim is to operate with and build on the existing USB ecosystem.

USB Power Delivery offers the following features:

- Increased power levels from existing USB standards up to 100W.
- Power direction is no longer fixed. This enables the product with the power (Host or Peripheral) to provide the power.
- Optimize power management across multiple peripherals by allowing each device to take only the power it requires, and to get more power when required for a given application.
- Intelligent and flexible system level management of power via optional hub communication with the PC.
- Allows low power cases such as headsets to negotiate for only the power they require.

Examples:

- Enables new higher power use cases such as USB bus powered Hard Disk Drives (HDDs) and printers. This eliminates the need for a separate power brick.
- A monitor with a supply from the wall can power, or charge, a laptop while still displaying.
- USB power bricks or chargers are able to supply power through a laptop's USB ports.
- Laptops and USB power bricks can provide higher power to battery powered devices (not currently defined by USB).
- Battery powered devices can get increased charging current from a hub and then give it back temporarily when the user's HDD requires to spin up.
- System level:



Power Delivery is designed to co-exist with standard USB Battery Charging implementations. Implementers should note that if they include battery charging capability in their devices or support for host adapters such as docks or ACAs they should also reference the Battery Charging Specification.

1.2 Objective of the Compliance Program

The benefits of a compliance program have been proven by the USB initiative: the proliferation of knowledge, more stringent testing, and a higher standard of quality. The purpose of the "Power Delivery Compliance" is to utilize the effectiveness of the USB-IF compliance program.

1.3 Scope of the Document

This document tests and/or checks for compliance with requirements specified in ***[PowerDelivery1.2]***.

1.4 Intended Audience

This specification is intended for developers of Hosts, Hubs, Peripherals and Cables which have support for Power Delivery capability.

1.5 Reference Documents

The following referenced documents can be found on the USB-IF website www.usb.org:

[PowerDelivery1.2]	*USB Power Delivery Specification Revision 1.1*
[OTG&EH2.0]	*On-The-Go and Embedded Host Supplement to the USB* 2.0 Specification, Revision 2.0 plus errata and ECR

[OTG&EH3.0]	*On-The-Go and Embedded Host Supplement to the USB* 3.0Specification, Revision 1.0 plus errata and ECR
[OTG&EH3.0ComplianceChecklist]	*OTG&EH 3.0 Compliance Checklist*
[OTG&EH2.0ComplianceChecklist]	*OTG&EH 2.0 Compliance Checklist*
[Micro-USB1.01]	*Universal Serial Bus Micro-USB Cables and Connectors Supplement to the USB 2.0 Specification,* revision 1.01.
[USB3.1]	*Universal Serial Bus Revision 3.1 including ECNs and errata.*
[USB2.0]	*Universal Serial Bus R*evision 2.0 Specification including ECNs and errata.
[USBType-C 1.0]	*USB Type-C Specification*
[USBSystemsChecklist]	*USB Compliance Checklist, Systems*
[USBPeripheralChecklist]	*USB Compliance Checklist, Peripheral (Excluding Hubs)*
[USBPeripheralSilicon]	*USB Compliance Checklist, Peripheral Silicon (Excluding Hub Silicon)*
[PDCommunicationsEngineMOI]	*Communications Engine PD Compliance MOI, developed by MQP Electronics Ltd.*
[PDDeterministicMOI]	*Deterministic PD Compliance MOI, developed by Ellisys*

2 ACRONYMS AND TERMS

This chapter lists and defines terms and abbreviations used throughout this specification. Other terms and abbreviations are provided in ***[PowerDelivery1.2]***.

Term	Description
Attached	USB Power Delivery ports which are mechanically joined with USB cable.
BIST	Built In Self Test – Power Delivery testing mechanism for the Phy Layer.
Cold Socket	A downstream port receptacle that does not apply vSafe5V on V_{BUS} until a plug insertion is detected.
Connected	USB Power Delivery ports which are actively communicating using the USB Power Delivery protocol.
Consumer	The capability of a PD Port (typically a Device's upstream port) to sink power from the power conductor (e.g. V_{BUS}).
Consumer/Provider	A Consumer with the additional capability to act as a Provider.
Contract	An agreement on both power level and direction reached between a Port Pair. A contract may be explicitly negotiated between the Port Pair or may be an implicit power level defined by the current state. While operating in Power Delivery mode there will always be either an explicit or implicit contract in place. The agreement may only be altered in the case of a negotiation, Hard Reset or failure of the Source.
Dead Battery	A device has a Dead Battery when the battery in a device is unable to power its functions.
Device	When lower cased (device), it refers to any USB product, either device or host.
Device Policy Manager	Module running in a Provider or Consumer that applies Local Policy to each port in the Device via the Policy Engine.
Downstream Port	Either a port in the Host or the ports defined in ***[USB2.0]***, **[USB3.1]** or Type-C as defined in ***[USBType-C 1.0]***. The default Host and Source.
Dual-Role Device	A product containing one or more Dual-Role Ports that are capable of operating as either a Source or a Sink.
Dual-Role Port	A Consumer/Provider or Provider/Consumer capable port that is a port capable of operating as either a Source or Sink.
HDD	A Hard Disk Drive.
Hard Reset	This is initiated by HardReset signaling from either Port Partner. It restores V_{BUS} to the default condition and resets the PD communications engine to its default state.
IR Drop	The voltage drop across the cable and connectors between the Source and the Sink. It is a function of the resistance of the ground wire in the cable plus the contact resistance in the connectors times the current flowing over the path.

Term	Description
Local Policy	Every Device has its own Policy, called the Local Policy, that is executed by its Policy Engine to control its power delivery behavior. The Local Policy at any given time may be the default policy, hard coded or modified by changes in operating parameters or one provided by the system Host or some combination of these. The Local Policy optionally may be changed by a System Policy Manager.
Message	The packet payload consisting of a header for control messages and a header and data for data messages as defined in Section 5.6.1.2.5 of [PowerDelivery1.1].
Messaging	Communication in the form of messages as defined in Chapter 5.9.9 of [PowerDelivery1.1].
Negotiation	This is the PD process whereby: 1. The Source advertises its capabilities. 2. The Sink requests one of the advertised capabilities. 3. The Source acknowledges the request and alters its output to satisfy the request. The result of the negotiation is a contract for power delivery/consumption between the Port Pair.
Packet	One entire unit of PD communication including a preamble, SOP*, payload, CRC and EOP as defined in Section 5.6 of [PowerDelivery1.1].
PD	USB Power Delivery
PD Capable	A port that supports USB Power Delivery.
PDUSB	USB Device Port or USB Host Port that is PD capable.
PD Connection	A Port Pair with an established contract.
Phy Layer	The Physical Layer responsible for sending and receiving messages across V_{BUS} between a Port Pair.
Policy	Policy defines the behavior of PD capable parts of the system and defines the capabilities it advertises, requests made to (re)negotiate power and the responses made to requests received.
Policy Engine	The Policy Engine interprets the Device Policy Manager's input in order to implement Policy for a given port and directs the Protocol Layer to send appropriate messages.
Port	An interface typically exposed through a receptacle, or via a plug on the end of a hard-wired captive cable. USB Power Delivery defines the interaction between a Port Pair.
Port Pair	Two attached PD Ports.
Port Partner	The USB Power Delivery contract is negotiated between a Port Pair connected by a USB cable. These ports are known as Port Partners.

Term	Description
Power Conductor	The wire delivering power from the Source to Sink. For example USB's V_{BUS}.
Power Consumer	See Consumer
Power Provider	See Provider
Protocol Error	An unexpected or unknown message, that cannot be handled by a given implementation.
Protocol Layer	The entity that forms the messages used to communicate information between Port Partners.
Provider	A capability of a PD Port (typically a Host, Hub, or Wall Wart DFP) to source power over the power conductor (e.g. V_{BUS}). This corresponds to a Type-A ro a Type-C Port with R_P asserted on its CC Wire.
Provider/Consumer	A Provider with the additional capability to act as a Consumer. This corresponds to a Dual-Role Type-A Port or a Dual-Role Type-C Port with R_P asserted on its CC Wire.
Reserve	Power which is kept back by a Provider in order to ensure that it can meet total power requirements of attached Consumers on at least one port.
Safe Operation	Sources must have the ability to tolerate 5V applied by both Port Partners.
Signalling	An ordered set of four K-codes used to indicate a particular line symbol e.g. Hard Reset as defined in Section 5.4 of [PowerDelivery1.1].
Sink	The port consuming power from V_{BUS}; most commonly a Device.
Soft Reset	A process that resets the PD communications engine to its default state.
Source	A role a port is currently taking to supply power over V_{BUS}; most commonly a Host or Hub downstream port.
System Policy	Overall system policy generated by the system, broken up into the policies required by each Port Pair to affect the system policy. It is programmatically fed to the individual Devices for consumption by their Policy Engines.
System Policy Manager	Module running on the USB Host. It applies the System Policy through communication with PD capable Consumers and Providers that are also connected to the Host via USB.
Tester	The Tester is assumed to be a piece of test equipment, or an assembly of pieces of test equipment, which manage(s) the testing process of a PD UUT.
Unit Under Test (UUT)	The PD device that is being tested by the Tester and responds to the initiation of a particular BIST test sequence.
Upstream Port	Typically a B port on a Device as defined in ***[USB2.0]***, ***[USB3.1]*** or Type-C Port as definied in ***[USBType-C 1.0]***. The default Device and Sink.
USB Powered State	Synonomous with the ***[USB2.0]*** and ***[USB3.1]*** definition of the powered state.
VI	Same as power (i.e. voltage * current = power)

Term	Description
Wall Wart	A power supply or “power brick” that is plugged into an AC outlet. It supplies DC power to power a device or charge a battery.

3 EXECUTIVE SUMMARY

This Power Delivery Compliance Plan is supplementary to the compliance programs operated by USB-IF to certify against ***[USB2.0]*** or ***[USB3.1]***; testing only new parameters and features defined in ***[PowerDelivery1.2]***. Parameters and features tested by the ***[USB2.0]*** or ***[USB3.1]*** compliance programs will not be tested here.

The significant features tested from *[PowerDelivery1.2]* are:

- Features of cables and connectors required for detection and higher power
- Power Delivery PHY level operation over V_{BUS}
- Power Supply operation and characteristics
- Signalling, messaging and protocol
- State operation and message sequences
- Local Policy relating to power management
- Interactions with System Policy Manager over USB and responses to System Policy Manager inputs.

The USB-IF Board reserves the right to re-certify products if, after USB-IF certification, the product is updated such that this adds a new capability which has not previously been tested.

4 SUBMISSION MATERIALS

4.1 VEND-INFO: Vendor Information File

4.1.1 Introduction

In order to expedite the testing process, the Vendor is required to provide a text file which defines in detail, the UUT to be tested. One such file is required for any significantly different configuration; for example a UUT which needs to demonstrate correct functionality with or without External Power would need two files; one with the ' Externally_Powered ' parameter defined as 'YES', and one with it defined as 'NO'.

The full test suite should then be run for each such file.

It is the intention that the testing is as automated as possible, so that interaction by the vendor during a test is not expected, or indeed encouraged.

The reason for this is that such details as Source or Sink Capabilities of the UUT should always match the Vendor Information File (VIF) during a given test sequence, thus allowing the vendor intentions to be checked against the UUT presented for test.

4.1.2 File Format

The file is a 'text' file which must use single byte characters from the ASCII character set or Windows Code Page 1252. The file defines a number of named parameters as 32 bit values, plus a small number of free text strings.

- Each entry is a Parameter Name string, followed by a ':', followed by a 32 bit value
- Each parameter must be defined on a separate row
- Each value is expressed in decimal, or 0xhexadecimal, or 'YES' or 'NO' (meaning 1 or 0 resp.)
- String values are allowed for vendor name, product name, version info and TID
- The Parameter Names all start with the character '$'
- a semicolon ';' and anything which follows on that row is a comment and will be ignored
- blank rows are permitted and are ignored
- inappropriate parameters for the UUT in question (e.g. cable parameters for a non-cable marker) may be omitted or left in place as place keepers

4.1.3 Example File with Added Explanation of Options Possible

```
$Vendor_Name: "Acme Co."

$Product_Name: "Gadget"

$Version_Info: "HW Vers 1.02"

$TID: "1234567890"     ; string (any length) as defined by USB-IF


; General ---------------------------------------------------

; for all UUTs


UUT_Device_Type: 1      ; 0 = Consumer Only (i.e., asserts Rd)

                        ; 1 = Consumer/Provider (i.e., asserts Rd at

                        ;     startup and can be directed to assert Rp)

                        ; 2 = Provider/Consumer (i.e., asserts Rp at
                        ;     startup and can be directed to assert Rd)

                        ; 3 = Provider Only (i.e., asserts Rp)

                        ; 4 = DRP (i.e., toggles between asserting Rp and Rd)

                        ; 5 = Cable (i.e., asserts Ra on VConn)


; Will it respond with GoodCRC to SOP* ?


SOP_Capable: YES               ; YES or NO

SOP_P_Capable: YES             ; YES or NO

SOP_PP_Capable: YES            ; YES or NO

SOP_P_Debug_Capable: NO                ; YES or NO

SOP_PP_Debug_Capable: NO       ; YES or NO


Type_C: YES                    ; YES or NO [PROBABLY NOT NEEDED]

Captive_Cable: NO                      ; YES or NO


; IF UUT IS CABLE MARKER

; ==================================================================================


; Structured VDM Content ----------------------------------------------

; Header ID information

Structured_VDM_Version: 1              ; 0, 1, 2 or 3
```

```
Data_Capable_as_USB_Host: YES ; YES or NO

Data_Capable_as_USB_Device: NO       ; YES or NO

Product_Type: 3                      ; must be 3 or 4

Modal_Operation_Supported: NO ; YES or NO

USB_VID: 0x1234                      ; 0xNNNN

; Cert Stat VDO

XID: 4294967295               ; (32 bit decimalidentifier issued by USB-IF)

; Product VDO

PID: 0x1234                   ; 0xNNNN

bcdDevice: 0x1234                    ; 0xNNNN

; Cable VDO

Cable_HW_Vers: 0x0                   ; 0xN

Cable_FW_Vers: 0x0                   ; 0xN

Type_C_to_Type_A_B_C: 2              ; 0, 1 or 2

Type_C_to_Plug_Receptacle: 0  ; 0 (plug) or 1 (receptacle)

Cable_Latency: 0x1            ; 0x1-0xa

                              ; 0x1 = <10ns

                              ; 0x2 = 10ns - 20ns

                              ; 0x3 = 20ns - 30ns

                              ; 0x4 = 30ns - 40ns

                              ; 0x5 = 40ns - 50ns

                              ; 0x6 = 50ns - 60ns

                              ; 0x7 = 60ns - 70ns

                              ; 0x8 = 60ns - 1000ns

                              ; 0x9 = 60ns - 2000ns

                              ; 0xa = 60ns - 3000ns

Cable_Termination_Type: 1     ; 0, 1, 2 or 3

                              ; 0 = Both ends Passive, VCONN not required

                              ; 1 = Both ends Passive, VCONN required

                              ; 2 = One end Active, one end passive, VCONN required

                              ; 3 = Both ends Active, VCONN required




Cable_SSTX1_Dir_Support: NO   ; YES or NO

Cable_SSTX2_Dir_Support: NO   ; YES or NO
```

```
Cable_SSRX1_Dir_Support: NO   ; YES or NO

Cable_SSRX2_Dir_Support: NO   ; YES or NO

Cable_VBUS_Current: 1         ; 0, 1or 2

                              ; 0 = 1.5A

                              ; 0 = 3A

                              ; 0 = 5A

VBUS_through_cable: YES               ; YES or NO

Cable_SOP''_controller: NO    ; YES or NO

Cable_Superspeed_Support:     ; 0 - 2

                              ; 0 = USB 2.0 only

                              ; 1 = USB 3.1 Gen 1

                              ; 2 = USB 3.1 Gen 2

; SVIDs Responder VDO

;;Cable_num_SVIDs: 96         ; decimal number

Cable_num_SVIDs_min: 96               ; decimal number

Cable_num_SVIDs_max: 96               ; decimal number



SVID_fixed: Yes                       ; is SVID list fixed *and* in order?


SVID1: 0x1234                 ; 0xNNNN

SVID1_num_modes_min: 3        ; 1 - 6

SVID1_num_modes_max: 3        ; 1 - 6


SVID1_modes_fixed: YES


SVID1_mode1_enter: Yes        ; for fixed positions of modes

SVID1_mode2_enter: Yes        ; for fixed positions of modes

SVID1_mode3_enter: No         ; for fixed positions of modes


SVID1_mode2_recog_mask: 0x0007f000    ; for dynamic mode allocation

SVID1_mode2_recog_value: 0x00035000   ; if recognised - try to enter mode


;(must try to enter any recognised modes and succeed in at least one)
```

```
; IF UUT IS NOT CABLE MARKER
; ================================================================================

; General for non-cable markers---------------------------------------------------
; is the UUT externally powered in the current compliance test
Externally_Powered: YES              ; YES or NO
USB_Comms_Capable: NO        ; YES or NO
DR_Swap_To_DFP_Supported: NO  ; YES or NO
DR_Swap_To_UFP_Supported: NO  ; YES or NO
VCONN_Swap_To_On_Supported: NO       ; YES or NO
VCONN_Swap_To_Off_Supported: NO      ; YES or NO
Responds_To_Discov_SOP: YES   ; YES or NO    [NEW]
Attempts_Discov_SOP: YES             ; YES or NO    [NEW]

; IF CAN BE SOURCE (UUT_Device_Type = 1, 2, 3 or 4)--------------------------------
; Source Capabilities ----------------------------------------
; (for UUTs able to behave as Providers)

Rp_Value: 2    ; 0=Default, 1=1.5A, 2=3A
Sends_Pings: NO                      ; YES or NO
PD_Power_As_Source: 3000                     ; in mW
USB_Suspend_May_Be_Cleared: YES      ; YES or NO



Num_Src_PDOs: 3                      ; 1 to 7
                              ; all voltages, powers and currents are
                              ; specified in the specification defined units

Src_PDO_Supply_Type1: 0              ; 0 Fixed
Src_PDO_Peak_Current1: 0             ; 0 to 3
Src_PDO_Voltage1: 100        ; must be 100 (5V)
Src_PDO_Max_Current1: 100            ; 0 to 500

                              ; if 2 or more Source PDOs...
Src_PDO_Supply_Type2: 0              ; 0, 1 or 2 Fixed,Battery or Variable
Src_PDO_Peak_Current2: 0             ; (if fixed) 0 to 3
```

```
Src_PDO_Voltage2:                     ; (if fixed)
Src_PDO_Max_Voltage2:         ; (if battery or variable)
Src_PDO_Min_Voltage2:         ; (if battery or variable)
Src_PDO_Max_Current2:         ; (if fixed or variable)
Src_PDO_Max_Power2:           ; (if battery)

                              ; if 3 or more Source PDOs...
Src_PDO_Supply_Type3: 0               ; 0, 1 or 2 Fixed,Battery or Variable
Src_PDO_Peak_Current3: 0              ; (if fixed) 0 to 3
Src_PDO_Voltage3:                     ; (if fixed)
Src_PDO_Max_Voltage3:         ; (if battery or variable)
Src_PDO_Min_Voltage3:         ; (if battery or variable)
Src_PDO_Max_Current3:         ; (if fixed or variable)
Src_PDO_Max_Power3:           ; (if battery)

                              ; if 4 or more Source PDOs...
Src_PDO_Supply_Type4: 0               ; 0, 1 or 2 Fixed,Battery or Variable
Src_PDO_Peak_Current4: 0              ; (if fixed) 0 to 3
Src_PDO_Voltage4:                     ; (if fixed)
Src_PDO_Max_Voltage4:         ; (if battery or variable)
Src_PDO_Min_Voltage4:         ; (if battery or variable)
Src_PDO_Max_Current4:         ; (if fixed or variable)
Src_PDO_Max_Power4:           ; (if battery)

                              ; if 5 or more Source PDOs...
Src_PDO_Supply_Type5: 0               ; 0, 1 or 2 Fixed,Battery or Variable
Src_PDO_Peak_Current5: 0              ; (if fixed) 0 to 3
Src_PDO_Voltage5:                     ; (if fixed)
Src_PDO_Max_Voltage5:         ; (if battery or variable)
Src_PDO_Min_Voltage5:         ; (if battery or variable)
Src_PDO_Max_Current5:         ; (if fixed or variable)
Src_PDO_Max_Power5:           ; (if battery)

                              ; if 6 or more Source PDOs...
Src_PDO_Supply_Type6: 0               ; 0, 1 or 2 Fixed,Battery or Variable
Src_PDO_Peak_Current6: 0              ; (if fixed) 0 to 3
```

```
Src_PDO_Voltage6:                     ; (if fixed)
Src_PDO_Max_Voltage6:         ; (if battery or variable)
Src_PDO_Min_Voltage6:         ; (if battery or variable)
Src_PDO_Max_Current6:         ; (if fixed or variable)
Src_PDO_Max_Power6:           ; (if battery)

                              ; if 7 Source PDOs...
Src_PDO_Supply_Type7: 0               ; 0, 1 or 2 Fixed,Battery or Variable
Src_PDO_Peak_Current7: 0              ; (if fixed) 0 to 3
Src_PDO_Voltage7:                     ; (if fixed)
Src_PDO_Max_Voltage7:         ; (if battery or variable)
Src_PDO_Min_Voltage7:         ; (if battery or variable)
Src_PDO_Max_Current7:         ; (if fixed or variable)
Src_PDO_Max_Power7:           ; (if battery)


; IF CAN BE SINK (UUT_Device_Type = 0, 1, 2 or 4) -----------------------------
; Sink Capabilities ----------------------------------------
; (for UUTs able to behave as Consumers)

PD_Power_As_Sink: 3000                ; in mW
Higher_Capability_Set: YES            ; YES or NO

Snk_PDO_Supply_Type1: 0               ; 0 Fixed
Snk_PDO_Voltage1:                     ; must be 100 (5V)
Snk_PDO_Op_Current1: 100              ; 0 to 500

Num_Snk_PDOs: 3                       ; 1 to 7

                              ; voltages, powers and currents are
                              ; specified in the spec units

                              ; if 2 or more Sink PDOs...
Snk_PDO_Supply_Type2: 0               ; 0, 1 or 2 Fixed,Battery or Variable
Snk_PDO_Voltage2:             ; (if fixed)
Snk_PDO_Max_Voltage2:         ; (if battery or variable)
```

```
Snk_PDO_Min_Voltage2:         ; (if battery or variable)
Snk_PDO_Op_Current2:          ; (if fixed or variable)
Snk_PDO_Op_Power2:            ; (if battery)

                              ; if 3 or more Sink PDOs...
Snk_PDO_Supply_Type3: 0              ; 0, 1 or 2 Fixed,Battery or Variable
Snk_PDO_Voltage3:             ; (if fixed)
Snk_PDO_Max_Voltage3:         ; (if battery or variable)
Snk_PDO_Min_Voltage3:         ; (if battery or variable)
Snk_PDO_Op_Current3:          ; (if fixed or variable)
Snk_PDO_Op_Power3:            ; (if battery)

                              ; if 4 or more Sink PDOs...
Snk_PDO_Supply_Type4: 0              ; 0, 1 or 2 Fixed,Battery or Variable
Snk_PDO_Voltage4:             ; (if fixed)
Snk_PDO_Max_Voltage4:         ; (if battery or variable)
Snk_PDO_Min_Voltage4:         ; (if battery or variable)
Snk_PDO_Op_Current4:          ; (if fixed or variable)
Snk_PDO_Op_Power4:            ; (if battery)

                              ; if 5 or more Sink PDOs...
Snk_PDO_Supply_Type5: 0              ; 0, 1 or 2 Fixed,Battery or Variable
Snk_PDO_Voltage5:             ; (if fixed)
Snk_PDO_Max_Voltage5:         ; (if battery or variable)
Snk_PDO_Min_Voltage5:         ; (if battery or variable)
Snk_PDO_Op_Current5:          ; (if fixed or variable)
Snk_PDO_Op_Power5:            ; (if battery)

                              ; if 6 or more Sink PDOs...
Snk_PDO_Supply_Type6: 0              ; 0, 1 or 2 Fixed,Battery or Variable
Snk_PDO_Voltage6:             ; (if fixed)
Snk_PDO_Max_Voltage6:         ; (if battery or variable)
Snk_PDO_Min_Voltage6:         ; (if battery or variable)
Snk_PDO_Op_Current6:          ; (if fixed or variable)
Snk_PDO_Op_Power6:            ; (if battery)
```

```
                              ; if 7 Sink PDOs...
Snk_PDO_Supply_Type7: 0              ; 0, 1 or 2 Fixed,Battery or Variable
Snk_PDO_Voltage7:             ; (if fixed)
Snk_PDO_Max_Voltage7:         ; (if battery or variable)
Snk_PDO_Min_Voltage7:         ; (if battery or variable)
Snk_PDO_Op_Current7:          ; (if fixed or variable)
Snk_PDO_Op_Power7:            ; (if battery)

; Request ----------------------------------------------------
; (for UUTs able to behave as Consumers)

No_USB_Suspend_May_Be_Set: YES              ; YES or NO
GiveBack_May_Be_Set: NO                     ; YES or NO

; IF CAN DO PR_SWAP (UUT_Device_Type = 1, 2 or 4) ----------------------------------

; Dual Role Devices ------------------------------------------

Accepts_PR_Swap_As_Src: NO           ; YES or NO
Accepts_PR_Swap_As_Snk: NO           ; YES or NO
Requests_PR_Swap_As_Src: NO          ; YES or NO
Requests_PR_Swap_As_Snk: NO          ; YES or NO

; IF UUT RESPONDS TO SOP DISCOVERY  -----------------------------------------------
; i.e. if Responds_To_Discov_SOP = YES

Structured_VDM_Version_SOP:  1              ; 0, 1, 2 or 3
Data_Capable_as_USB_Host_SOP: YES           ; YES or NO
Data_Capable_as_USB_Device_SOP: YES  ; YES or NO
Product_Type_SOP: 2                  ;  must NOT be 3 or 4
Modal_Operation_Supported_SOP: YES          ; YES or NO
USB_VID_SOP: 0x1234                  ; 0xNNNN
; Cert Stat VDO
XID_SOP: 4294967295           ; (32 bit decimal identifier issued by USB-IF)
; Product VDO
PID_SOP:                             ; 0xNNNN
```

```
bcdDevice_SOP: 0x1234                ; 0xNNNN


Num_SVIDs_min_SOP: 96         ; decimal number

Num_SVIDs_max_SOP: 96         ; decimal number

SVID_fixed_SOP: YES           ; YES or NO


SVID1_SOP: 0x1234                    ; 0xNNNN

SVID1_num_modes_min_SOP: 3    ; 1 - 6

SVID1_num_modes_max_SOP: 3    ; 1 - 6


SVID1_modes_fixed_SOP: YES


SVID1_mode1_enter_SOP: Yes           ; for fixed positions of modes

SVID1_mode2_enter_SOP: Yes           ; for fixed positions of modes

SVID1_mode3_enter_SOP: No            ; for fixed positions of modes


SVID1_mode2_recog_mask_SOP: 0x0007f000       ; for dynamic mode allocation

SVID1_mode2_recog_value_SOP: 0x00035000      ; if recognised - try to enter mode


;(must try to enter any recognised modes and succeed in at least one)


; IF UUT RESPONDS TO SOP DISCOVERY and Product_Type_SOP is 5 (AMA) ----------------------

; AMA VDO


AMA_HW_Vers: 0x0              ; 0xN

AMA_FW_Vers: 0x0              ; 0xN

AMA_SSTX1_Dir_Support: NO     ; YES or NO

AMA_SSTX2_Dir_Support: NO     ; YES or NO

AMA_SSRX1_Dir_Support: NO     ; YES or NO

AMA_SSRX2_Dir_Support: NO     ; YES or NO

AMA_VCONN_power: 2            ; 0, 1, 2, 3, 4, 5, 6

                              ; 0 = 1W

                              ; 1 = 1.5W

                              ; 2 = 2W

                              ; 3 = 3W

                              ; 4 = 4W
```

```
                              ; 5 = 5W
                              ; 6 = 6W
AMA_VCONN_reqd:        YES             ; YES, NO
AMA_VBUS_reqd: YES            ; YES, NO
AMA_Superspeed_Support:       ; 0 - 3
                              ; 0 = USB 2.0 only
                              ; 1 = USB 3.1 Gen 1
                              ; 2 = USB 3.1 Gen 2
                              ; 3 = USB 2.0 Billboard Only
```

4.1.4 Vendor File FAQs

UUT_Device_Type

Options are:

0 - Consumer Only (i.e asserts Rd all the time)

1 - Consumer/Provider (i.e., asserts Rd at startup and can be directed to assert Rp)

2 - Provider/Consumer (i.e., asserts Rp at startup and can be directed to assert Rd)

3 - Provider Only (i.e., asserts Rp)

4 - DRP (i.e., toggles between asserting Rp and Rd)

5 - Cable (i.e., asserts Ra on VConn, communicates using SOP' and SOP'')

*An AMA, for example, is any of the options **except** cable, however much it may look like a cable.*

USB_VID

Field name for VID when UUT is a cable marker (communicates with SOP’ or SOP”)

USB_VID_SOP

Field name for VID when UUT communicates with SOP.

SOP_Capable, SOP_P_Capable, SOP_PP_Capable, SOP_P_Debug_Capable, SOP_PP_Debug_Capable.

The requirement is to answer 'YES' if the UUT will return a GoodCRC for a message in the appropriate SOP* type, under the conditions of the test, when it is appropriate.
Typically SOP_P_Capable and SOP_PP_Capable will not be responded to by a UFP with a contract. During testing this is the assumed condition. If they respond to SOP' and/or SOP'' as a DFP then the question should be answered 'YES'.

Externally_Powered USB_Comms_Capable DR_Swap_Capable

Each of these represents a bits in more than one message. It is important that all messages give the same value for each of these bits.

Externally Powered

This means that the UUT currently has available, power from a wall outlet or AC adapter. Large internal batteries do not count as external power. (More information exchange about batteries is the subject of future specfication development.)

PD_Power_as_Source

An ECR to the V2.0 PD Specification has established a new set of Power Rules, superceding the earlier Profile Rules. The vendor of a Source Capable device shall specify a PD_Power level (PDP) in mW. This is used by the Tester in conjunction with the Normative Voltages and Currents Table to determine the minimum capabilities which may be offered by the Source.

PD_Power_as_Sink

The vendor of a Sink Capable device shall specify a PD_Power level (PDP) in mW. This is used by the Tester in conjunction with the Normative Voltages and Currents Table to determine the maximum capabilities which may be required by the Sink.

Source Capabilities offered by Tester

Please note that the first Source Capabilities message sent by the tester to a Sink will always be 5V at 100mA. The UUT is required to make a legal request for this (or for less current). At the same time the Capability Mismatch bit may be set. Any UUT requesting more than the Tester offered Source Capabilities will automatically fail, and no further testing will be performed.

At at least one point a Sink UUT will be offered 0mA at all voltages offered. This is a PD Suspend condition and a request for 0mA at one of the voltages offered is a mandatory requirement. Any UUT which does not request 0mA under these conditions will fail compliance.

No_USB_Suspend_May_Be_Set

USB_Suspend_May_Be_Cleared

In a *Request* message there is a bit called *No USB Suspend,* which the Sink sets to 1 if it wishes to request that it need not obey USB Suspend, but instead be allowed to continue to draw power according to its PD contract.

In the Source Capabilities there is a bit called *USB Suspend Supported*, which is normally set to 1, but may be cleared to zero to signal to the sink that the sink need not obey USB Suspend.

If the UUT (as a Sink) will ever set *No USB Suspend* to a 1, then the VIF must set No_USB_Suspend_May_Be_Set to 'YES'. Otherwise it should be set to 'NO'.

If the UUT (as a Source) will ever set *USB Suspend Supported* to a 0, then the VIF must set `USB_Suspend_May_Be_Cleared to` 'YES'. Otherwise it should be set to 'NO'.

At the moment (in the current state of testing) the safest value for these two VIF parameters is YES, as it allows both 1 and 0 for the related bits to be accepted as valid. It is recommended however that the correct value be entered as explained above, in order not to fall foul of more sophisticated testing which may be introduced. An incorrect statement may affect the Compliance validity.

GiveBack_May_Be_Set

In a *Request* message there is a bit called *Giveback,* which the Sink sets to 1 if it is prepared to lower its Operating Current to its Minimum Operating Current, on demand.

If the UUT will ever set this bit in the test configuration, then it must set **`GiveBack_May_Be_Set`** to'YES'. Otherwise it should be set to 'NO'.

Higher_Capability_Set

In a *Sink Capabilities* message there is a bit called *HigherCapability,* which the Sink sets to 1 if it needs more than ***vSafe5V*** (e.g. 12V) to provide full functionality.

If the UUT sets this bit in the configuration described by the VIF, then Higher_Capability_Set mus be set to 'YES', otherwise, to 'NO'.

PD Compliance Testing of Modes

Full testing of SVIDs and Modes is outside the scope of the PD Compliance Plan. It depends on other specification, in some cases proprietary to other organisations.

Anticipated Flexibility

The list of SVIDs announced by the device in response to a Discover SVIDs initiator will be either

- A fixed list of SVIDs in a fixed sequence, defined in advance by the vendor. or
- A variable length list of SVIDs in no particular sequence, but always selected from a comprehensive master list which the vendor is able to define in advance. Such a list must always contain at least one SVID.

For each of these SVIDs there will be (in response to a Discover Modes initiator) either

- A fixed list of Modes in a fixed sequence, defined in advance by the vendor. At least one Mode must obey Enter and Exit commands. or
- A variable length list of Modes in no particular sequence, but always containing one or more Modes which can be uniquely identified from a specified substring in the mode, and which will obey Enter and Exit commands.

Details of Vendor Supplied Information and Testing

The following sections describe the parameter names used for SVIDs and Modes, and describing whether a given mode should be tested for the ability to Enter and Exit a given mode.

SVID1 - SVID100
SVID1_SOP - SVID100_SOP

SVIDs discovered from cables are referred to by a name such as *SVID1*, up to *SVID100*. (For practical reasons only 100 SVIDs may be defined in the test.)

SVIDs discovered using SOP are referred to by a name such as *SVID1_SOP*, up to *SVID100_SOP*. (For practical reasons only 100 SVIDs may be defined in the test.)

Cable_num_SVIDs_min, Num_SVIDs_min_SOP
Cable_num_SVIDs_max, Num_SVIDs_max_SOP

These define the smallest and largest number of SVIDs which the UUT will ever announce in response to *Discover SVIDs*.

SVID_fixed, SVID_fixed_SOP

If the *SVID_fixed* parameter is defined as 'YES', then *Cable_num_SVIDs_min* must equal *Cable_num_SVIDs_max,* and the test will be that exactly those SVIDs are announced and in that exact sequence.

If the *SVID_fixed* parameter is defined as 'NO', then *Cable_num_SVIDs_min* (*Num_SVIDs_min_SOP*) must be less than or equal to *Cable_num_SVIDs_max (Num_SVIDs_max_SOP*)*,* and the test will be that those SVIDs which are announced are all included in the predefined *SVID1* etc (*SVID1_SOP* etc) values. There must be at least one SVID announced. The number of *SVID1* etc (*SVID1_SOP* etc) values defined must be at least *Cable_num_SVIDs_max* (*Num_SVIDs_max_SOP*)*.*

Modes for each SVIDx

Modes are defined for each SVID. Two situations are allowed for:

- the modes for a given SVID are a fixed number (between 1 and 6) and have a fixed sequence in the Discover Modes ACK message (so can always be recognised by the Tester by the position number), or
- the number of modes is variable and the mode is recognisable by a 32 bit mask and value.

Modes are referred to by a prefix such as *SVID1_mode1* to *SVID1_mode100*. (For practical reasons only 100 Modes per SVID may be defined in the test.) We do not specifically require the mode values themselves.

SVID1_modes_fixed (YES), etc
SVID1_mode1_enter, etc

If the list of Modes for a given SVID is fixed, then the *SVID1_mode1_enter* parameters are defined.

The test will then be that exactly those Modes are announced and in that exact sequence. In this case the number of modes for this SVID is limited to must lie between 1 and 6. If the associated parameter SVID1_mode1_enter is defined as YES, then it must be possible to Enter and Exit the mode.

SVID1_modes_fixed (NO), etc
SVID1_mode2_recog_mask, etc
SVID1_mode2_recog_value, etc

If the list of Modes for a given SVID is variable, then (e.g.) *SVID1_mode2_recog_mask and SVID1_mode2_recog_value* are defined for Modes for which Enter and Exit must work. Each such parameter pair must identify a unique mode. Each such mode must be able to be entered, and at least one such mode must be present for any given SVID. The mode numbers given here must start at 1 and have contiguous numbering up to the largest given.

SVID1_num_modes_min
SVID1_num_modes_max

The value of *SVID1_num_modes_min is the smallest number of modes which will ever be announced for this SVID,* the value of *SVID1_num_modes _max is the largest number of modes which will ever be announced for this SVID. So SVID1_num_modes _max can exceed the number of SVID1_mode2_recog_mask and SVID1_mode2_recog_value parameters, because only the modes which can be Entered are included in the latter list.*

5 TEST CONDITIONS

5.1 Relationship to Legacy USB Testing

A USB capable UUT being tested for PD is also required to pass the appropriate USB 2.0/USB 3.1 testing. In the case where the UUT is dependent on PD for its successful USB operation, the USB 2.0/USB 3.1 test equipment will be required to offer sufficient PD functionality for this to take place.

5.1.1 Assertions Covered

6.4.1.3.2#1 Prior to a PD contract, [the USB Suspend Supported] flag is undefined and Sinks shall follow the rules for suspend as defined in ***[USB2.0]****,* **[USB3.1]** *or* **[BC1.2]**.

Include standard tests as a pre-requisite on the checklist.

6.4.1.3.2#2 If the USB Suspend Supported flag is set, then the Sink shall follow the ***[USB2.0]*** *or* **[USB3.1]** *rules for suspend and resume but may draw up to pPDSuspend.*

Need USB protocol and a current measurement.

6.4.1.3.2#3 If the USB Suspend Supported flag is cleared, then the Sink shall ignore the **[USB2.0]** *or* **[USB3.1]** *rules for suspend and may continue to draw the negotiated power. However, when USB is suspended, the USB device state is also suspended.*

Need USB protocol and a current measurement.

5.2 Temperature

It is assumed that all measurements for Compliance will be taken at room temperature. No special provision is expected to be made for any extended temperature range.

5.3 Point of Compliance for Signal Measurements

The point of compliance is at the edge of the UUT connector. In practice, it is impossible to achieve this without introducing some error, however simulations have been performed which suggest that sufficiently accurate results can be achieve using the following techniques:

- The test cable should be kept to a sufficient short length, approximately 250mm maximum.
- Care should be taken that the cable VBUS characteristic impedance closely matches the termination impedance in the Tester.
- The point of application of the measurement probe should be designed not to have any significant effect on the transmission line properties of the VBUS connection, by avoiding significantly low impedances and stubs.
- The equipment should be calibrated, and the measurements made should be adjusted to

take into account the effects of the cable.

Provided these measures are taken, measurements taken with the test equipment can easily meet the expected requirements.

This statement does not preclude the option of using a break-out board near the UUT end of the cable, as the measurement point. It is a matter for the test equipment manufacturer to be able to demonstrate the suitability of the test equipment offered.

To summarize:

- The point of measurement is the edge of the UUT connector, so all measurements will need to be compensated/corrected/calibrated to allow for the effects of the measurement equipment connection.

- Note that for a UUT with a captive cable, the point of compliance is at the plug of this cable. However as this captive cable is assumed to be the only cable between the UUT and any other port, in certain test different transmit and receive level limits are tested for, such that the levels tested for are equivalent to to those which would have been seen at the Tester end of the cable had the cable not been a captive one.

5.4 BMC-ALG-CLK-RECOV: Clock Recovery for BMC Transmitter Mask Measurements

Some potential problems arise when constructing an eye diagram:

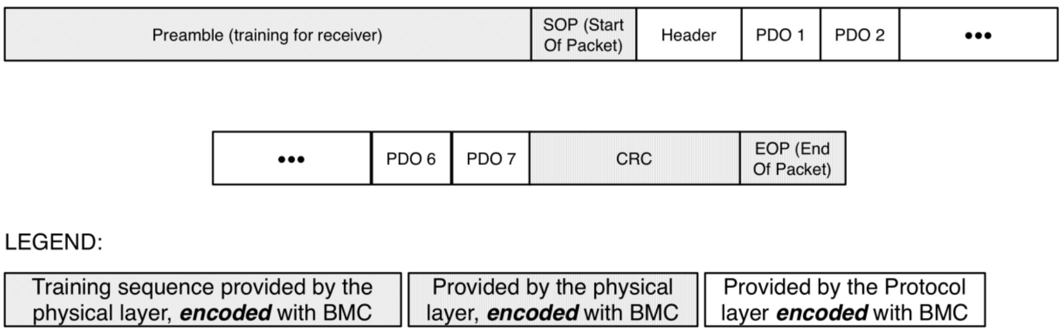
- In order to perform transmitter mask measurement the data clock is required for correct display of the waveform.
- The bit rate is allowed a certain amount of drift within a packet, so the clock period is not fixed.

5.4.1 Clock Recovery

In an ideal situation the clock used for the display of the eye diagram would originate at the transmitter (in the Unit under Test, or UUT) and would be connected directly to the eye diagram display equipment. However it is a characteristic of the PD UUT that this signal is not available, and that all tests take place on the accessible CC connection which passes between units.

Therefore we need to use a clock recovered from the data. The algorithm used for clock recovery must be defined so that test equipment from different vendors generate an equivalent recovered clock.

The eye diagram should be constructed, and the bit rates measured, using data runs equivalent to the longest packet used in PD, which is the packet sent in the BIST Test Pattern.



This has a total of 64 + 20 + 20 + 7*40 + 40 + 5 = 429 bits.

5.4.2 Reference Bit Rate

For each packet transmitted the PD specification defines the reference bit rate to be the average bit rate of the last 32 bits of the preamble. This is the bit rate against which any bit rate drift is measured.

5.4.3 Clock and Bit Rate Recovery Algorithm

It is intended that the technique used be a reasonable model of what a practical receiver may achieve in a real-time environment. The method described below employs the reference bit rate, but regularly adjusts the clock phase to allow a symmetrical eye diagram representation. The value UI from the specification is defined here by the reference bit rate.

Measurements of pBitRate and fBitRate are made separately in other BMC-PHY chapter tests.

5.4.4 Reference Bit Rate measurement algorithm

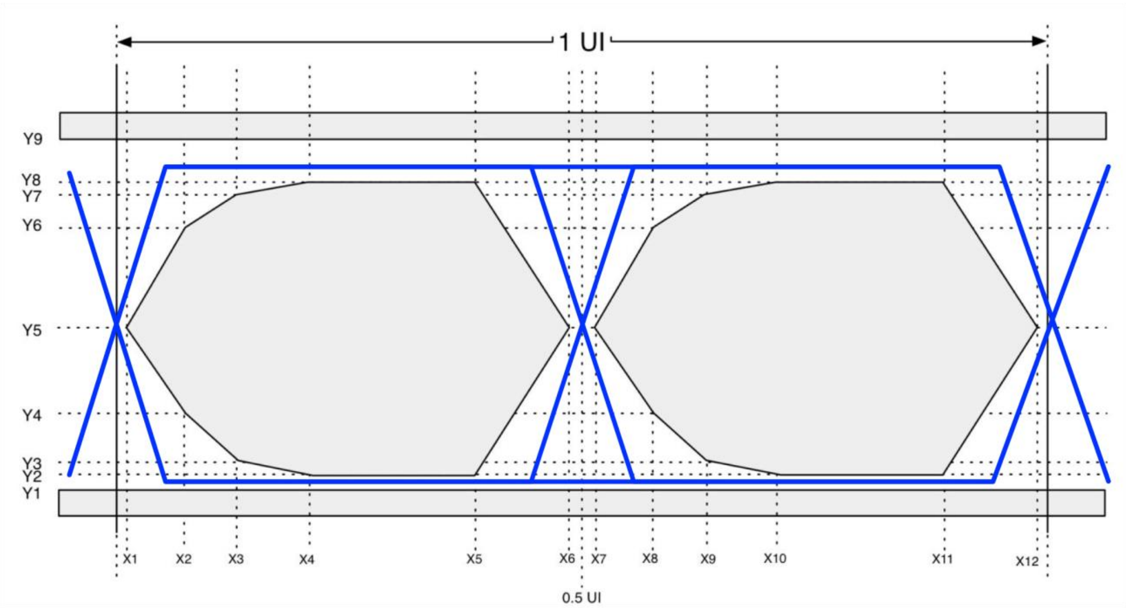
The proposed method of determining the reference bit rate is as follows:

- Choose a sequence of 32 bits starting at a zero bit.
- Fit a fixed rate clock to these bits, using a least mean squares technique on the zero crossing errors over this number of bits.
- This defines the reference bit rate of the recovered clock at the start of the following 365 bits.

5.4.5 Clock recovery algorithm

The method of recovering the clock is as follows:

- Take transitions corresponding the bit boundaries, and plot the data waveform around each bit centre position onto the mask. Plot both 0's and 1's on to the same eye diagram to ensure that if they have different lengths, this will be observed.



5.5 Eye Diagram

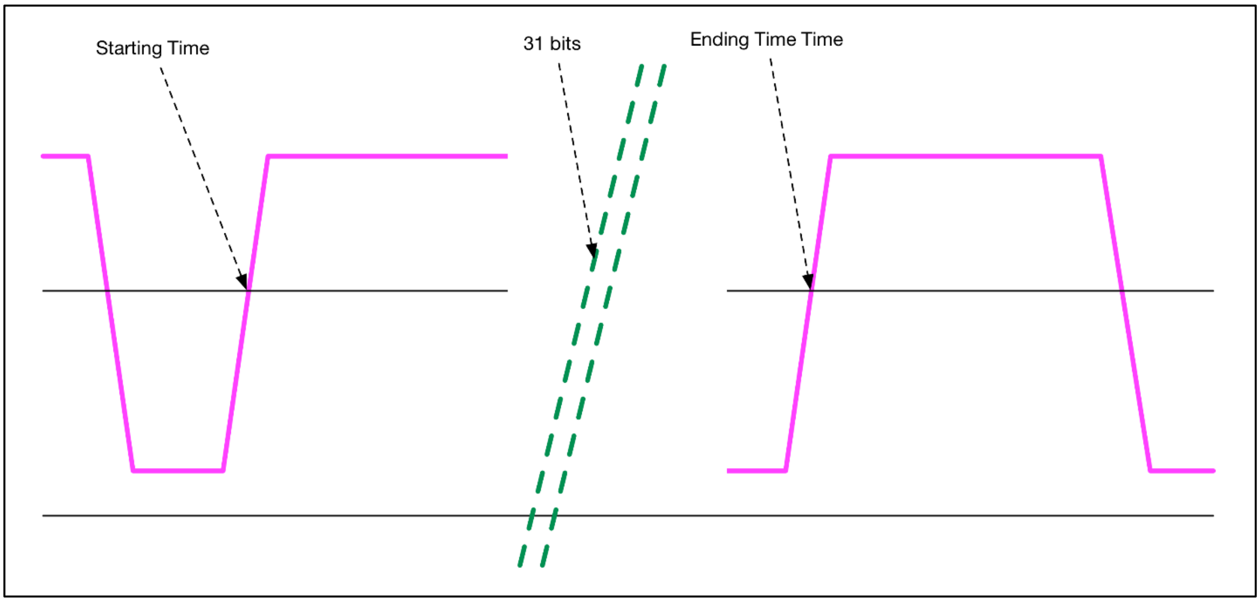
5.5.1 Introduction

This is the procedure for drawing the BMC EYE-Diagram.

5.5.2 Reference Frequency

Measure the time over 32 bits. Use the following procedure:

1. Locate the rising edge of a "0" bit. This is the start of the measured time.
2. Count 32 bits.
3. Locate the rising edge of a "0" bit. This is the end of the measured time.
4. Divide the measured time by 32, this is now the reference bit length = 1 UI.



5.5.3 Drawing of the EYE Diagram

The EYE Diagram shall use 2640 bits placed on the TX-Mask.

The TX Mask is drawn using the reference frequency measured in the previous step.

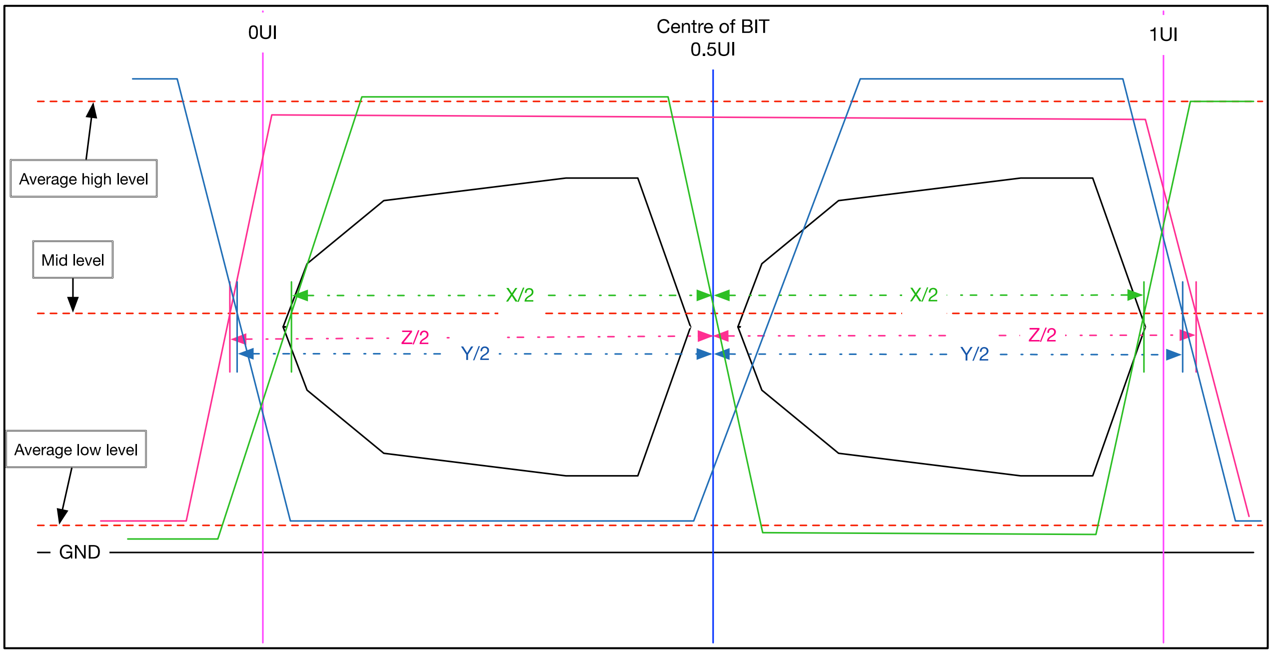
Each bit is treated individually and is drawn symmetrically from the centre of the bit. This means that any deviation from the ideal bit length will be distributed evenly at both ends of the bit.

The initial crossing point is set to the midpoint between the average of the high levels for all bits and the average of the low levels for all bits.

If the resulting eye diagram does not pass, the crossing point level is shifted as follows and the eye diagram is re-evaluated for each step below until it passes:

1. Crossing point at the midlevel (first eye to be shown)
2. Add 10mv to the initial crossing point level
3. Subtract 10mv from the initial crossing point level
4. Add 20mv to the initial crossing point level
5. Subtract 20mv from the initial crossing point level
6. Add 30mv to the initial crossing point level
7. Subtract 30mv from the initial crossing point level
8. Add 40mv to the initial crossing point level
9. Subtract 40mv from the initial crossing point level
10. Add 50mv to the initial crossing point level
11. Subtract 50mv from the initial crossing point level

If the eye diagram does not pass with any of these settings, it has failed.



5.6 VBUS Voltage Measurement

The previous content of this section related specifically to the FSK version of PD. A new section will be written to replace it.

5.7 VBUS Current Measurement

A number of parameters involving the measurement of current, are defined in the PD Specification. In cases where these are defined as maximum current values, the actual method specified in compliance will be to average the measured value over 250ms, and compare this with the maximum parameter value specified. This is to bring the measurement into line with the way in which typical measuring instruments function. In practice any averaging period from 100ms to 500ms will be acceptable.

For measurements involving overshoots (such as in 6.4.1.2.2.5 Peak Current) the current will be measured at the output of a first order filter with a time constant of 200µs.

5.8 PDO Transition Patterns

When testing a Source, it is sometimes necessary that the Tester should switch between PDOs in a Capabilities message in such a way as to confirm the result of every possible transition. An example of this is when performing voltage increase/decrease tests. It is desirable that the minimum number of transitions is performed to reduce testing time and avoid repeat a transition.

The following are possible patterns of transitions which allow all transitions to be tested for a given number of PDOs in the minimum number of steps. Other sequences are possible.

Number of PDOs in Capabilities Message						
1	**2**	**3**	**4**	**5**	**6**	**7**
1 to 1*	1 to 2	1 to 2	1 to 2	1 to 2	1 to 2	1 to 2
	2 to 1	2 to 1	2 to 1	2 to 1	2 to 1	2 to 1
		1 to 3	1 to 3	1 to 3	1 to 3	1 to 3
		3 to 2	3 to 2	3 to 2	3 to 2	3 to 2
		2 to 3	2 to 3	2 to 3	2 to 3	2 to 3
		3 to 1	3 to 1	3 to 1	3 to 1	3 to 1
			1 to 4	1 to 4	1 to 4	1 to 4
			4 to 3	4 to 3	4 to 3	4 to 3
			3 to 4	3 to 4	3 to 4	3 to 4
			4 to 2	4 to 2	4 to 2	4 to 2
			2 to 4	2 to 4	2 to 4	2 to 4
			4 to 1	4 to 1	4 to 1	4 to 1
				1 to 5	1 to 5	1 to 5
				5 to 4	5 to 4	5 to 4
				4 to 5	4 to 5	4 to 5
				5 to 3	5 to 3	5 to 3
				3 to 5	3 to 5	3 to 5
				5 to 2	5 to 2	5 to 2
				2 to 5	2 to 5	2 to 5
				5 to 1	5 to 1	5 to 1
					1 to 6	1 to 6
					6 to 5	6 to 5
					5 to 6	5 to 6
					6 to 4	6 to 4
					4 to 6	4 to 6
					6 to 3	6 to 3

					3 to 6	3 to 6
					6 to 2	6 to 2
					2 to 6	2 to 6
					6 to 1	6 to 1
						1 to 7
						7 to 6
						6 to 7
						7 to 5
						5 to 7
						7 to 4
						4 to 7
						7 to 3
						3 to 7
						7 to 2
						2 to 7
						7 to 1

* A transition is possible from any PDO to the same PDO, having its own timing constraints.

5.9 PROT-CALC-HR: Inactivity to Hard Reset Delay Calculation

5.9.1 Source

Some tests require to know how long, after a break in communication, a Hard Reset should be expected to be intiated by a Source. The calculation follows:

- A Ping must be sent after a maximum of tSourceActivity max (50000 µs) of any inactivity by the Source.
- The Ping message itself contains 149 data bits and lasts for a maximum of 149/270 ms = 552 µs.
- A retry must be sent within tReceive max (900 µs) + tRetry max (75 µs) = 975 µs
- The first retry lasts a maximum of 552 µs.
- The next retry must be sent within 975 µs

- The second retry lasts a maximum of 552 µs.
- The Hard Reset must be sent within tReceive max (900 µs) + tHardReset max (5000 µs) = 5900 µs
- The Hard Reset itself contains 84 data bits and lasts for a maximum of 84/270 ms = 311 µs.
- So the maximum permitted time before receiving Hard Reset is
  50000 + 552 + 975 + 552 + 975 + 552 + 5900 + 311 = 59817 µs

  For the sake of simplicity we round this up to **60 ms.**

5.9.2 Sink

For a Sink, the calculation is much simpler, being tSinkActivity max = **150 ms**.

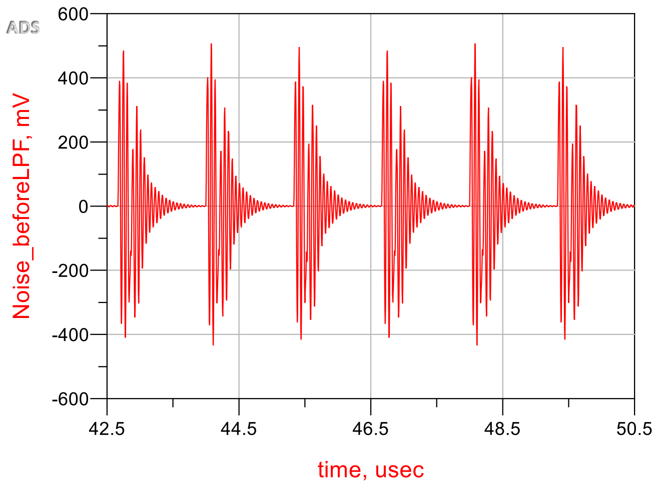
5.10 Interference Rejection – Tester Electrical Requirement

1. Source impedance of the test transmitter from source and noise generators shall be
   - 50 Ω ± 5 Ω
   - 400 pF ± 50 pF
2. Noise shall be generated from one of the three approaches described below.
3. The dynamic range of the noise amplitude at the input of UUT shall be less than 1V.

5.10.1 BMC Noise Generation Requirement

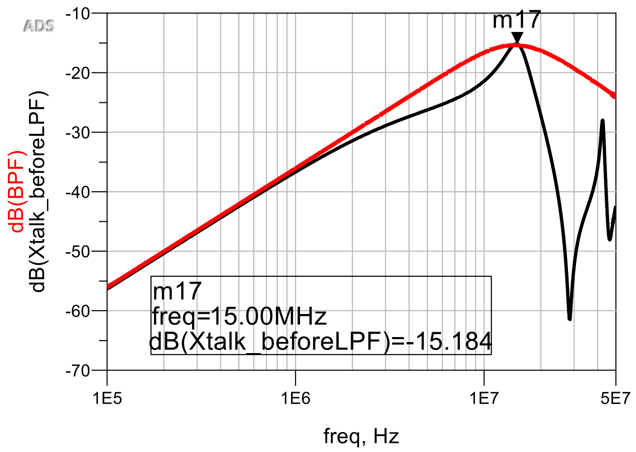
The three approaches are listed in this document for noise implementation guidance and flexibility. The noise magnitude shall be calibrated to reach the maximum level of vNoiseActive, 165mV. The noise calibration is done by connecting the test transmitter with 200pF shunt capacitive load in series with the Rx bandwidth limiting filter with the time constant tRxFilter. The Rx bandwith limiting filter shall be implemented with at least 5 kΩ load impedance to achieve a time constant of tRxFilter. The scope capacitance loading effect shall be taken into account.

1. Directly from the waveform text file using an arbitrary waveform generator (AWG). The time step shall be ≤ 15 nsec. The peak-to-peak noise magnitude shall be calibrated to 165 mV in the scope. (Note: Refer to USB.org for the waveform text file)



USB FullSpeed SE0 Xtalk Noise Waveform at the Input of UUT

2. Using a band pass filter (BPF) circuitry with the peak frequency at 15 ± 1 MHz, 20dB/dec slope before the peak frequency and -20dB/dec slope after the peak frequency. The noise waveform at the input of the BPF has the pulse width 167 ns and the pulse period 1.33us. The peak-to-peak noise magnitude shall be calibrated to 165 mV in the scope. (Note: Refer to USB.org for the reference schematic of the BPF)



Transfer Functions of USB FullSpeed SE0 to CC Xtalk (black solid line)
and Band Pass Filter (red dash line)

3. Using two-tone sinusoidal noise waveforms simultaneously. The in-band peak-to-peak noise at 750 ± 50 kHz and the out-of-band peak-to-peak noise at 5 ± 0.5 MHz shall be calibrated to 90mV and 75 mV in the scope, respectively.

   When any of the above three approaches is used, the noise phase relative to the signal shall be advanced through the signal period so that the worst case of the interference situation can be captured.

6 TEST SETUP

The tests in this section test only a partial list of all the possible parameters and compliant behavior. The tests should not be considered as a full validation test plan. It is the responsibility of the manufacturer of a device to verify compliance of their products according to xxxx.

6.1 Test Set Ups

6.2 User Input Before Test Runs

Before running any test suite, the Tester needs to be informed of a number of parameters by the test operator. This information will be available from the Vendor Information File (VIF).

6.3 Pass Criteria

In the some of the test sequences which follow, a particular form of wording has been used, to ensure that the pass criteria are clear. Wherever the word 'check' is used, this defines a timing or behavior requirement that must be satisfied for the overall test sequence to be deemed to have been passed. The failure to satisfy any one of these 'checks' results in a failure for the test sequence in question.

For example, in the following test sequence fragment, two pass criteria are implicitly defined by use of the word 'check'. Failure of either one results in a failure for the complete test sequence.

...

7. Check for V_{BUS} on within T_{PWRUP_RDY} (30s)

8. Check it remains on for $T_{A\_WAIT\_BCON}$ min (1.1s)

...

6.4 Parameter v Test Identifier

Section 1 identifies which test procedure(s) result in each of the parameters specified in ***[PowerDelivery1.2]*** being tested. Not all parameters can be directly measured.

7 ASSERTIONS

7.1 Chapter 1 Introduction

7.1.1 Normative (Shall/Normative)

R2.0 Assertion #	Description	Test	Comments
SECTION 1 Introduction			
Subsection 1.4 Conventions			
1.4.1 Precedence			
1.4.1#1	If there is a conflict between text, figures, and tables, the precedence shall be tables, figures, and then text.		
1.4.2.3 Discarded			
1.4.2.3#1	Discarded is a keyword indicating that a packet, when received, shall be thrown away by the PHY Layer and not passed to the Protocol Layer for processing. No GoodCRC Message shall be sent in response to the Packet.		
1.4.2.4 Ignored			
1.4.2.4#1	Ignored is a keyword indicating messages or message fields which, when received, shall result in no action by the receiver, aside from returning a GoodCRC message to acknowledge message receipt		
1.4.2.6 N/A			
1.4.2.6#1	N/A is a keyword that indicates that a field or value is not applicable and has no defined value and shall not be checked or used by the recipient.		
1.4.2.7 Optional/Optionally/Optional Normative			
1.4.2.7#1	if an optional feature is implemented, the feature shall be implemented as defined by this specification (optional normative).		
1.4.2.8 Reserved			
1.4.2.8#1	Their use and interpretation may be specified by future extensions to this specification and shall not be utilized or adapted by vendor implementation.		
1.4.2.8#2	A reserved bit, byte, word, or field shall be set to zero by the sender…		
1.4.2.8#3	A reserved bit, byte, word, or field...shall be ignored by the receiver.		
1.4.2.8#4	Reserved field values shall not be sent by the sender…		
1.4.2.8#5	Reserved field values...shall be ignored by the receiver.		

7.2 Chapter 3 Type A and Type B Cable Assemblies and Connectors Assertions

7.2.1 Normative (Shall/Normative)

R2.0 Assertion #	**Description**	**Test**	**Comments**
SECTION 3 Cable Assemblies and Connectors			
Subsection 3.1 Significant Features			
Subsection 3.1.1 Connectors			
3.1.1#1	PD Micro_A, PD Micro_AB, and PD Micro_B connectors shall be capable of carrying 3A current as specified in Section 3.6.5.1.		
3.1.1#2	PD Standard_A and PD Standard_B connectors shall be capable of carrying 5A current as specified in Section 3.6.5.2.		
3.1.2 Compliant Cable Assemblies			
3.1.2#1	PD cable assemblies shall have one plug on each end.		
3.1.2#2	Any cable combinations not explicitly defined in the list above shall not be permitted.		
3.1.2#3	Connectors on each end of the PD cable assembly shall be labeled with the appropriate USB PD icon defined in Section 3.5.		
3.1.4 Hardwired Captive PD Cable			
3.1.4#1	The connector on the hardwired captive PD cable assembly shall be labeled with the appropriate USB PD icon defined in Section 3.5.		
3.1.5 Standard_A Insertion Detect			
3.1.5#1	The Insertion Detect feature shall be implemented for cold socket Standard_A applications		
3.1.6 Standard_A PD Detect			
3.1.6#1	Implementation of PD Detect shall include either pin 10, pin 11, or both pin 10 and pin 11 in the PD Standard_A receptacle.		
3.1.6#2	If both pins 10 and 11 are present then both shall be connected to the PD Detect logic.		
3.1.6#3	PD Detect shall be an open circuit when PD is not detected (refer to Figure 3_2, Figure 3_3 and Figure 3_4)		
3.1.6#4	PD Detect...shall be connected to the receptacle shield when a PD Standard_A plug is present (Figure 3_5).		
3.1.6#5	PD Detect shall be a closed circuit to the shield when the PD plug is detected.		
3.1.6#6	The PD Standard_A receptacle shall not connect PD Detect to the receptacle shield when a USB Thin Card is inserted, as shown in Figure 3_4.		
3.2.1 Standard_A Insertion Detect Mechanical Dimensions			
3.2.1#1	Figure 3_6 shows the mechanical dimensions for the zone where the Insertion Detect mechanism on the Standard_A and PD Standard_A receptacle shall activate when any type of Standard_A plug is inserted		
3.2.1#2	The detection mechanism shall be designed such that the insertion detect electrical requirements are met, regardless of power being applied.		
3.2.2 USB PD Standard_A PD Detect Mechanical Requirement			
3.2.2#1	Figure 3_7 shows the mechanical dimensions for the range where the PD Detect mechanism on the PD Standard_A receptacle shall indicate PD Detect when a PD Standard_A plug is inserted.		
3.2.2#2	The dimensions shall apply to planes parallel to Datum A		
3.2.2#3	The dimensions...shall apply to both USB 2.0 and USB 3.1 versions of PD Standard_A receptacles.		
3.2.2#4	The implementation shall also conform to the following requirements: · The metal shell of the USB PD Standard_A plug shall be gold plated on the inner and outer surfaces a minimum of 1.6 mm		

R2.0 Assertion #	Description	Test	Comments
	from the leading edge.		
3.2.2#5	The implementation shall also conform to the following requirements: · Contact for PD Detect shall occur on either the inner or the outer surface of the plug shell.		
3.2.2#6	The implementation shall also conform to the following requirements: · The insertion of a USB ThinCard shall not indicate PD Detect.		
3.2.3.1 Interface Definition (USB 2.0 PD Standard_A Connector)			
3.2.3.1#1	Figure 3_8 defines the mechanical requirements that govern the mating interoperability that shall be followed for the USB 2.0 Standard_A plug.		
3.2.3.1#2	When designing stacked USB 2.0 PD Standard_A receptacles, PD Detect contacts shall be provided in all PD capable receptacles.		
3.2.3.2 Pin Assignments and Description (USB 2.0 PD Standard_A Connector)			
3.2.3.2#1	The usage and assignments of the pins that shall be used in the USB 2.0 PD Standard_A connector are defined in Table 3_2.		
3.2.3.2#2	Note 2: Implementation of PD DETECT shall include: a) either pin 10 or pin 11. b) both pin 10 and pin 11.		
3.2.3.2#3	Note 3: Pin 12, if present, shall be connected to Shield.		
3.2.4.1 Interface Definition (USB 3.1 PD Standard_A Connector)			
3.2.4.1#1	Figure 3 11 defines the mechanical requirements of the USB 3.1 PD Standard_A plug that govern the mating interoperability that are different than the dimensions in the [USB2.0]and [USB3.1] specifications, that shall be followed, for the USB 3.1 Standard_A plug.		
3.2.4.1#2	The USB 3.1 PD Standard_A receptacle mating interface shall comply with the dimensions defined in the [USB2.0] and [USB3.1] specifications for the USB 3.1 Standard_A receptacle.		
3.2.4.1#3	When designing a stacked USB 3.1 PD Standard_A receptacle, PD Detect contacts shall be provided in all PD capable receptacles.		
3.2.4.2 Pin Assignments and Descriptions (USB 3.0 PD Standard_A Connector)			
3.2.4.2#1	The usage and assignments of the pins that shall be used in the USB 3.1 PD Standard_A connector are defined in Table 3_3.		
3.2.4.2#2	Note 3: Implementation of PD DETECT shall include : a) either pin 10 or pin 11. b) both pin 10 and pin 11.		
3.2.4.2#2	Note 4: Pin 12, if present, shall be connected to Shield.		
3.2.5.1 Interface Definition (USB 2.0 PD Standard_B Connector)			
3.2.5.1#1	The USB 2.0 PD Standard_B plug and receptacle shall comply with the mechanical requirements specified in the [USB2.0] base specification		
3.2.5.1#2	The USB 2.0 PD Standard_B plug and receptacle shall comply with...the ID pin as specified in Figure 3 15 and Figure 3 14.		
3.2.5.2 Pin Assignments and Descriptions (USB 2.0 PD Standard_B Connector)			
3.2.5.2#1	The usage and assignments of the pins that shall be used in the USB 2.0 PD Standard_B connector are defined in Table 3 4.		
3.2.6.1 Interface Definition (USB 3.1PD Standard_B Connector)			
3.2.6.1#1	The USB 3.1 PD Standard_B plug and receptacle shall comply with all other mechanical requirements specified in the [USB3.1] base specification		
3.2.6.1#2	The USB 3.1 PD Standard_B plug and receptacle shall comply with...the ID pin as specified in Figure 3 17 and Figure 3 18.		
3.2.6.2 Pin Assignments and Descriptions (USB 3.1 PD Standard_B Connector)			
3.2.6.2#1	The usage and assignments of the pins that shall be used in the USB 3.1 PD Standard_B connector are defined in Table 3 5.		

R2.0 Assertion #	Description	Test	Comments
3.3 Cable Assemblies			
3.3#1	The B_Device shall detect the cable assembly and make requests that are consistent with the cable assembly's capabilities.		
3.3#2	The portion of this specification that allows the negotiation of voltages other than the default vSafe5V and currents in excess of 1.5A shall apply only to Devices with marked PD cable assemblies.		
3.3.1 Non_marked Cable Assemblies			
3.3.1#1	Devices shall only use these non_marked legacy cable assemblies at vSafe5V and up to 1.5A as described by [USB2.0], [USB3.1] and [BC1.2].		
3.3.2 Marked Cable Assemblies			
3.3.2#1	Port Partners shall not negotiate for currents in excess of the electrical characteristics indicated by the cable's marking.	BMC_PROT_SEQ_CHKCAB_P_PC_2	
3.4.1 Marker for PD Standard_A Connectors			
3.4.1#1	The PD Standard_A Connector shell shall be detected by the DFP connector using a PD Standard_A receptacle connector and associated marking detection circuitry to indicate that the cable assembly is a PD cable assembly.		
3.4.2 Electronic Markers for Micro_A Plugs			
3.4.2#1	The data and shield connections shall be made per the [USB2.0] and the [USB3.1] specifications.		
3.4.2.1 Legacy Micro_A Plug			
3.4.2.1#1	For PD to remain backward compatible, the low impedance to ground shall be maintained.		
3.4.2.3 PD Micro_A plug			
3.4.2.3#1	PD cable assemblies with Micro_A plug connectors shall be marked with rID terminating the ID pin to ground.		
3.4.3 Electronic Markers for PD Standard_B Plugs and Micro_B Plugs			
3.4.3#1	Note the 3A and 5A markers are mutually exclusive; hence both markers shall not be present at the same time.		
3.4.3.1 3A Capable PD B Plug			
3.4.3.1#1	The schematic diagram for a 3A_capable PD cable assembly detailing how a PD Standard_B plug connector or a Micro_B plug connector shall be marked is shown in Figure 3 24.		
3.4.3.2 5A Capable PD B Plug			
3.4.3.2#1	The schematic diagram for a 5A_capable PD cable assembly detailing how a PD Standard_B plug connector shall be marked is shown in Figure 3 25.		
3.5 USB PD Icon			
3.5#1	The USB PD cable assemblies shall display the USB icon appropriate for PD.		
3.6 USB Power Delivery Cable Requirement			
3.6#1	The USB PD connector family shall conform to all the requirements in Section 3, in addition to the requirements specified in the [USB2.0], [USB3.1] or [BC1.2] specifications.		
3.6#2	Test sequences shall conform to EIA_364_1000.001 as specified in the Environmental Requirements section of [USB3.1].		
3.6.1 Low Level Contact Resistance (EIA 364_23B)			
3.6.1#1	The power contacts of a 3A PD cable assembly shall conform to the following requirements: • 20mΩ (Max) initial for VBUS and GND contacts. • Maximum change (delta) of +10mΩ after environmental stresses. • Measure at 20mV (Max) open circuit at 100mA.		
3.6.1#2	The power contacts of a 5A PD cable assembly shall conform to the following requirements:		

R2.0 Assertion #	Description	Test	Comments
	• 20mΩ (Max) initial for VBUS and GND contacts. • Maximum change (delta) of +10mΩ after environmental stresses. • Measure at 20mV (Max) open circuit at 100mA.		
3.6.1#3	The following requirements shall apply, independent of power being applied, to the resistance of a PD Standard_A plug receptacle shell to the PD Standard_A receptacle PD DETECT contact with a PD Standard_A plug in the mated condition or to the resistance between the INSERTION DETECT contacts when a Standard_A plug is present in the Standard_A receptacle supporting Insertion Detect. • 200mΩ (Max) initial. • Maximum change (delta) of 300mΩ after environmental stresses. • Measure at 20mV (Max) open circuit at 100mA		
3.6.2 Open Circuit Resistance			
3.6.2#1	The following requirements shall apply, independent of power being applied, to the resistance of the PD Standard_A receptacle shell to the PD Standard_A receptacle PD DETECT contact(s) when no PD Standard_A plug is inserted in the mated condition or a non_PD Standard–A plug is inserted in the mated condition or to the resistance between the INSERTION DETECT contacts of a Standard_A receptacle supporting Insertion Detect when a Standard_A plug is not inserted: • ≥ 10MΩ initial. • ≥ 10MΩ after environmental stress		
3.6.3 Dielectric Strength (EIA 364_20)			
3.6.3#1	No breakdown shall occur when 100VAC (RMS) is applied between adjacent contacts of unmated and mated connectors.		
3.6.4 Insulation Resistance (EIA 364_21)			
3.6.4#1	There shall be a minimum of 100MΩ insulation resistance between adjacent contacts of unmated and mated connectors.		
3.6.5.1 3A PD Connector Mated Pair (EIA 364_70, Method 2)			
3.6.5.1#1	When a current of 3.0 A is applied to the VBUS pin and its corresponding GND pin (i.e., pin 1 and pin 5 of the PD Micro_A Connector, PD Micro_AB Connector, or PD Micro_B Connector), the delta temperature shall not exceed +30°C at any point on the connectors under test, when measured at an ambient temperature of 25°C.		
3.6.5.2 5A PD Connector Mated Pair (EIA 364_70, Method 2)			
3.6.5.2#1	When a current of 5.0A is applied to the VBUS pin and its corresponding GND pin (i.e., pin 1 and pin 4 of the PD Standard_B Connector or PD Standard_A Connector), the delta temperature shall not exceed +30°C at any point on the connectors under test, when measured at an ambient temperature of 25°C.		
3.6.6 Differential Crosstalk between VBUS and the D+/D_ Pair (EIA_360_90)			
3.6.6#1	The differential, near_end, and far_end, crosstalk between the D+/D_ pair and VBUS shall be managed not to exceed the limit shown in Figure 3_26.		
3.6.7 PD Cable Assembly Shielding Connectivity			
3.6.7#1	The shield conductor shall be attached to the connector shell at both ends of the cable and shall provide a resistance of no greater than 1.0Ω from end to end.		
3.6.7#2	The shield shall not be connected to ground within the cable.		
3.6.8 PD Cable VBUS Impedance			
3.6.8#1	The cable impedance shall meet the requirements specified in Table 5_18.		
3.6.9 PD Cable Insertion Loss			

R2.0 Assertion #	Description	Test	Comments
3.6.9#1	The insertion loss of the cable shall not exceed aInsertionLoss.		
3.6.10.1 Voltage Drop on the Ground			
3.6.10.1#1	A PD cable assembly shall ensure VGND_DROP does not exceed the maximum value listed in Table 3_7 for the signal ground reference between the Source's power source connection to the USB receptacle and the Sink's connection to its internal power conversion block, if present. VGND_DROP shall be measured at iCable and measured either: · At the receptacle's solder pad where it attaches to the printed circuit board · Or where the captive cable is attached to the printed circuit board in the device.		
3.6.10.1#2	VGND_DROP shall include the effects of the following: a) The mated contact resistance of both the A_side and the B_side (if present) connections for GND. b) The series resistance of the cable's GND wire. c) The rated current of the cable (iCable).		
3.6.10.1#3	The ground wire size shall be selected so that VGND_DROP at a current of iCable does not exceed the maximum value listed in Table 3_7		
3.6.10.2 Overall IR Drop between a Source and a Sink			
3.6.10.2#1	A PD cable assembly shall provide a maximum voltage drop of vIRDrop_Cable for VBUS to GND at the Sink end of the cable.		
3.6.10.2#2	The VBUS to GND voltage drop shall include VVBUS_DROP plus VGND_DROP .		
3.6.10.2#3	VVBUS_DROP shall be measured at a current of iCable and measured either: · At the receptacle's solder pad where it attaches to the printed circuit board · Or where the captive cable is attached to the printed circuit board in the device.		
3.6.10.2#4	VVBUS_DROP shall include the effects of the following · The mated contact resistance of both the A_side and B_side (if present) connections for VBUS. · The series resistance of cable's VBUS wire. · The rated current of the cable (iCable).		
3.6.10.2#5	The VBUS wire size shall be selected so that VVBUS_DROP at a current of iCable does not exceed the maximum value listed in Table 3_7.		
3.7 Electrical Parameters			
3.7#1	For charging_only cables (e.g. cables without data lines), the maximum value of either VVBUS_DROP or VGND_DROP may be exceeded; however VVBUS_DROP plus VGND_DROP shall not exceed vIRDrop_Cable.		

7.2.2 Recommended (Should)

R2.0 Assertion #	Description	Test	Comments
SECTION 3 Cable Assemblies and Connectors			
3.1.7 Raw Cables			
3.1.7#r1	Attention should be given for the possibility of signal interference from VBUS, especially at higher operating current and voltage, in PD cable assemblies that are capable of Hi_Speed or SuperSpeed data communication.		
3.2.1 Standard_A Insertion Detect Mechanical Dimensions			
3.2.1#1	Special consideration should be given to the location of shell features such as retention springs and cutout areas.		
3.3.1 Non_marked Cable Assemblies			
3.3.3#1	PD cable assemblies using these larger limits may result in mechanical interferences (e.g., may block use of a connector slot		

R2.0 Assertion #	**Description**	**Test**	**Comments**
	in stacked connector configurations) therefore, the overmold dimensions should conform to requirements defined by [USB2.0] and [USB3.1] if possible.		

7.3 Chapter 4 Electrical Requirements

7.3.1 Normative (Shall/Normative)

R2.0 Assertion #	Description	Test	Comments
SECTION 4 Cable Assemblies and Connectors			
4#1	In the case where a Device requests power via the Battery Charging Specification and then the USB Power Delivery Specification, it shall follow the USB Power Delivery Specification until the port pair is unattached or there is a hard reset.		
4#2	If the USB Power Delivery connection is lost, the port shall return to its default state, see Section 6.7.2.		
4.1.1.1 Overview			
4.1.1.1#1	A Consumer/Provider shall back power the Provider/Consumer at a reduced current level.	POW_DB_CP	
4.1.1.2 Consumer/Provider Operational Details			
4.1.1.2#1	In operation (see Figure 4_1), a Consumer/Provider port that does not detect vSafe5V on VBUS shall periodically apply a current limited five volt supply (vSafeDB) to VBUS in an attempt to ascertain the presence of a Provider/Consumer port that wants to be powered.	POW_DB_CP	
4.1.1.2#2	In response to vSafeDB on VBUS, a Provider/Consumer port that wants power shall transmit the Bit Stream that a Consumer/Provider can detect indicating the presence of a port that wants to be powered.	POW_DB_CP	
4.1.1.2#3	When the Consumer/Provider detects a Provider/Consumer port wanting to be powered, it shall apply vSafe5V to VBUS.	POW_DB_CP	
4.1.1.2#4	In all other cases, for example when connected to a legacy port or an unpowered Provider port, there will be no response and the Consumer/Provider port shall not apply full power (vSafe5V) to VBUS.	POW_DB_CP	
4.1.1.2#5	The Consumer/Provider shall continue to periodically probe for the presence of a Provider/Consumer port that wants power.	POW_DB_CP	
4.1.1.2#6	The limitations on the current and power applied to back power VBUS and its controlled duration shall be applied in order to prevent damage to legacy ports.	POW_DB_CP	
4.1.1.3 Provider/Consumer Operational Details			
4.1.1.3#1	In operation (see Figure 4_1), the Provider/Consumer with a dead battery shall begin (as it must if its battery is truly dead) by outputting nothing on VBUS.	POW_DB_PC	
4.1.1.3#2	The Provider/Consumer shall begin outputting the Bit Stream on VBUS within tSendBitStream.	POW_DB_PC	
4.1.1.1#3	When this logic is ready, the Provider/Consumer port shall signal its Port Partner that is ready to operate as Consumer by stopping the Bit Stream.	POW_DB_PC	
4.1.1.4 Sequence of Operation			
4.1.1.4#1	To ensure consistent behavior all Consumer/Provider ports shall have the ability to detect a Provider/Consumer with a Dead Battery or Unpowered Port.		
4.1.1.4#2	The Provider/Consumer shall not drive VBUS; allowing it to remain at vSafe0V.		
4.1.1.4#3	The Consumer/Provider shall start DBDetectTimer, used to determine when next to apply vSafeDB.		
4.1.1.4#4	While vSafeDB is not present, the Provider/Consumer shall: • If willing to act as a Source, go and do this. (The first action will be to output vSafe5V). • If it does not wish to be powered, remain in this step.		
4.1.1.4#5	Until the DBDetectTimer expires, the Consumer/Provider shall check whether VBUS is above vSafe0V. If it is, it shall start to operate as a Sink, and leave the Dead Battery		

R2.0 Assertion #	Description	Test	Comments
	Operation procedure.		
4.1.1.4#6	When the DBDetectTimer expires, the Consumer/Provider shall start to output vSafeDB, in order to power the Provider/Consumer Bit Stream generation circuitry.		
4.1.1.4#7	The Consumer/Provider shall then start the BitStreamDetectTimer, which determines how long to wait for the Bit Stream.		
4.1.1.4#8	The Bit Stream detector shall see at least 128 alternating '0' and '1' bits before deciding that the Bit Stream is present.		
4.1.1.4#9	If the BitStreamDetectTimer expires before the Bit Stream is detected, the Consumer/Provider shall perform the following procedure.		
4.1.1.4#9a	The Consumer/Provider shall attempt to discharge the maximum permitted capacitance (this can theoretically be as much as cSrcBulk max or cSrcBulkShared max) on VBUS at a rate which will reduce its voltage to vSafe0V within tDBDischargeVbus max bearing in mind that VBUS may be being driven with vSafe5V and not capable of being discharged.		
4.1.1.4#9b	After tDBDischargeVbus max the Consumer/Provider shall stop attempting to discharge VBUS.		
4.1.1.4#9c	After a time of tDBSourceOff from starting to discharge VBUS, then the Consumer/Provider shall go back to step 1.		
4.1.1.4#10	If the Provider/Consumer wishes to be powered, and vSafeDB is present, it shall start to output Bit Stream within tSendBitStream of vSafeDB being available, and start the WaitForPowerTimer.		
4.1.1.4#11	When budgeting for tSendBitStream the Provider/Consumer shall allow for: • The time required for vSafeDB to charge the capacitance presented by itself to a level at which it can begin operation. • The time required for enough of its own circuitry to power up to transmit a Bit Stream. • The fact that tTurnOnImpliedSink is included in this budget		
4.1.1.4#12	If the Bit Stream is detected, the Consumer/Provider shall output vSafe5V to VBUS and start the DeviceReadyTimer.		
4.1.1.4#13	When the WaitForPowerTimer expires, the Provider/Consumer assumes that vSafe5V is present on VBUS. It shall then bring up the full PD system and when ready to receive and process Source Capabilities messages, stop sending the Bit Stream.		
4.1.1.4#14	If the DeviceReadyTimer expires, the Consumer/Provider shall perform the Apply vSafe0V Procedure defined in step 4 above, ending by going to step 1.		
4.1.1.4#15	When the Source Capabilities message is received, the Provider/Consumer shall determine the connected cable type, and start operation as a Sink.		
4.1.1.4#16	If the Bit Stream is detected to have stopped, the Consumer/Provider shall go and start operating as a Source. The first action will be to send the Source Capabilities message.		
4.1.1.4#17	Note: If at any time the Provider/Consumer sees vSafe0V on VBUS, it shall ensure that it is not sending Bit Stream within tBitStreamOff, and go to step 1.		
4.1.1.4 Sequence of Operation (SUMMARIZED)			
4.1.1.4#1	To ensure consistent behavior all Consumer/Provider ports shall have the ability to detect a Provider/Consumer with a Dead Battery or Unpowered Port.	POW_DB_CP	
4.1.1.4#2	In summary the Provider/Consumer that requires power shall:	POW_DB_PC	

R2.0 Assertion #	Description	Test	Comments
	1. Discharge VBUS to vSafe0V.		
4.1.1.4#3	In summary the Provider/Consumer that requires power shall: 2. When a voltage greater than vSafe0V is detected on VBUS, start sending the Bit Stream within tSendBitStream using vSafeDB to power its PD interface.	POW_DB_PC	
4.1.1.4#4	In summary the Provider/Consumer that requires power shall: 3. After sending the Bit Stream for tWaitForPower, assume vSafe5V is present on VBUS. (Note there is no positive handshake for this transition.)	POW_DB_PC	
4.1.1.4#5	In summary the Provider/Consumer that requires power shall: 4. Use vSafe5V to bring up its Protocol and Policy engines and when it's ready to receive and process Source Capabilities messages stop sending the Bit Stream.	POW_DB_PC	
4.1.1.4#6	In summary the Provider/Consumer that requires power shall: 5. Operate as Consumer awaiting receipt of a Source Capabilities message	POW_DB_PC	
4.1.1.4#7	The Consumer/Provider, when it detects VBUS is at vSafe0V for tDBDetect, shall: 1. Output vSafeDB for at least tBitStreamDetect min and for no more than tBitStreamDetect max.	POW_DB_CP	
4.1.1.4#8	The Consumer/Provider, when it detects VBUS is at vSafe0V for tDBDetect, shall: 2. Detect at least 128 alternating '0s' and '1s' from the Bit Stream o If the Bit Stream is not detected, remove vSafeDBfrom VBUS, wait tDBDetect and go to step 1. o Output vSafe5V and start DeviceReadyTimer.	POW_DB_CP	
4.1.1.4#9	The Consumer/Provider, when it detects VBUS is at vSafe0V for tDBDetect, shall: 3. While the Bit Stream is present o tDeviceReady not exceeded, go to 4. o tDeviceReady exceeded, remove vSafe5V from VBUS, wait tDBDetect and go to step 1.	POW_DB_CP	
4.1.1.4#10	The Consumer/Provider, when it detects VBUS is at vSafe0V for tDBDetect, shall: 4. Send a Source Capabilities message and operate as a Provider.	POW_DB_CP	
4.3 A_Plug Insertion Detect			
4.3#1	When a plug is present for cold socket, applications shall be used to indicate when to apply power to Vbus.	not tested	
4.3#2	The Insertion Detect feature for Standard_A receptacles shall be present for cold socket	not tested	
4.4.1 Detecting Cabling Capabilities			
4.4.1#1	Only A_plug to B_plug assemblies that are marked for PD shall be used for voltages higher than vSafe5V or current levels higher than 1.5A.		
4.4.1#2	The Source shall limit maximum capabilities it offers so as not to exceed to the capabilities of the type of plug detected.		
4.4.1#3	Requests made by the Sink shall not exceed the capabilities of the type of plug.		
4.4.1#4	Sources shall run the cable detection process prior to the Source sending Source Capabilities messages offering voltages in excess of vSafe5V or currents in excess of: _1.5A for Type_A/Type_B _3A for Type_C		
4.4.1#5	Sinks with Type_A and Type_B connectors shall run the cable		

R2.0 Assertion #	Description	Test	Comments
	detection process prior to sending any Request Messages.		
4.4.1#6	Sinks with Type_C connectors shall select Capabilities from the offered Source Capabilities assuming that the Source has already determined the Capabilities of the cable.		
4.4.1#7	Provider/Consumers with dead batteries shall wait until after receipt of the first Source Capabilities message before running the cable detection process and making a request for power.		
4.4.1#8	Sources shall detect the type of attached cable and either limit the Capabilities they offer or operate in a Low Power mode based on: · The receptacle type (i.e. Standard, Micro or C_Type) and its known current carrying capability. · The current carrying capability of the plug determined by: o The plug type (i.e. Type_C, PD, non_PD or Low Power). o Cable capabilities determined using Structured VDM messages (see Section 6.4.4.2) sent using SOP' Communication (see Section 2.4).		
4.4.1#9	Sinks, except those with Type_C receptacles, shall detect the type of attached cable and limit their requests based on: · The receptacle type (i.e. Standard, or Micro) and its known current carrying capability. · The current carrying capability of the plug determined by: o The plug type (i.e. PD or non_PD).		
4.4.2 Plug Type Determination			
4.4.2#1	Figure 4_2 shows the flow for the first portion of the multi_step process that shall be used to determine the kind of cable.		
4.4.3 Detecting the PD Capabilities of the Standard_A Connector			
4.4.3#1	Figure 4_3 illustrates how the detection contacts in the A receptacle shall be used.		
4.5 Low Power Devices using Micro_A Plug			
4.5#1	Sources that are able to detect the Low Power plug shall automatically begin an Implicit Contract equivalent to a Battery PDO (Max Voltage = vLowPower max, Min Voltage = vLowPower min, Max Power=pLowPower nom) and shall not transmit or respond to PD messaging or signaling while the Low Power plug is connected.	not tested	
4.5#2	Low Power Devices (Sinks) shall be able to operate normally when powered by any voltage in the range vLowPower and shall not transmit USB PD messaging or signaling.	not tested	
4.5#3	The Sink shall be able to operate on a voltage on the range vLowPower and shall draw no more than pLowPower nominal.	not tested	
4.5#4	Sources that are not able to detect the Low Power plug shall treat the plug as a PD micro_A plug.	not tested	

7.3.2 Recommended (Should)

R2.0 Assertion #	Description	Test	Comments
SECTION 4 Electrical Requirements			
Subsection 4.2 Cable IR Ground Drop			
4.2#r1	Certified USB cableing is specdified such that errors should not occur (See Section 3.6.10)	not tested	
Subsection 4.3 Plug Insertion Detect			
4.3#r1	The Insertion Detect feature : • Should be used to Iindicate to the PD logic to start sending Source Capabilities messages when a plug is present.	not tested	
4.3#r2	The Insertion Detect feature : • Should be used to Iindicate to the PD logic to put VBUS back	not tested	

R2.0 Assertion #	**Description**	**Test**	**Comments**
	to the default state when the plug is removed.		

7.4 Chapter 5 Physical Layer

7.4.1 Normative (Shall/Normative)

R2.0 Assertion #	Description	Test	Comments
SECTION 5 Physical Layer			
Subsection 5.3 Symbol Encoding			
5.3#1	Except for the preamble, all communications on the line shall be encoded with a line code to ensure a reasonable level of DC_balance and a suitable number of transitions.	BMC_PHY_MSG_6	
5.3#2	4b5b line code shall be used.	BMC_PHY_MSG_6	
5.4 Ordered sets			
5.4#1	Ordered sets shall be interpreted according to Figure 5_1.	PHYFSK_HR	
5.4#2	The receiver shall search for all four K_codes and when it finds at least three in the correct place, it may interpret it as a valid ordered set (see Table 5_3).		
5.5 Transmitted Bit Ordering			
5.5#1	This section describes the order of bits on the wire that shall be used when transmitting data of varying sizes.	BMC_PHY_MSG_6	
5.5#2	Figure 5_2 shows the transmission order that shall be followed.	BMC_PHY_MSG_6	
5.6 Packet Format			
5.6#1	The packet format shall consist of a Preamble, an SOP*, (see Section 5.6.1.2), packet data including the Message header, a CRC and an EOP (see Section 5.6.1.5).	BMC_PHY_MSG_6	
5.6#2	The packet format is shown in Figure 5_3 and indicates which parts of the packet shall be 4b/5b encoded.	BMC_PHY_MSG	
5.6#3	Once 4b/5b encoded, the entire Packet shall be transmitted either using BFSK over VBUS or BMC over CC. Note that when using BMC the Preamble is BMC encoded.	BMC_PHY_MSG	
5.6.1.1 Preamble			
5.6.1.1#2	Unlike the rest of the packet, the preamble shall not be 4b/5b encoded.	BMC_PHY_MSG_6	
5.6.1.1#3	The preamble shall consist of a 64_bit sequence of alternating 0s and 1s.	BMC_PHY_MSG_6	
5.6.1.1#4	The preamble shall start with a "0" and shall end with a "1".	BMC_PHY_MSG_6	
5.6.1.2.1 Start of Packet Sequence (SOP)			
5.6.1.2.1#1	A Power Delivery Capable Provider, Provider/Consumer, Consumer or Consumer/Provider shall be able to detect and communicate with packets using SOP .	BMC_PHY_MSG_1	
5.6.1.2.1#2	If a valid SOP is not detected (see Table 5_3) then the whole transmission shall be Discarded.	BMC_PHY_MSG_2	
5.6.1.2.1#3	Sending and receiving of SOP Packets shall be limited to PD Capable DFPs and UFPs only (i.e. PD Capable Ports on PDUSB Hosts and PDUSB Devices).	CBL_PHY_MSG	
5.6.1.2.1#4	Cable Plugs shall neither send nor receive SOP Packets.	CBL_PHY_MSG	
5.6.1.2.1#5	Note that PDUSB Devices, even if they have the physical form of a cable (e.g. AMAs), are still required to respond to SOP Packets.	CBL_PHY_MSG	
5.6.1.2.2 Start of Packet Sequence Prime (SOP')			
5.6.1.2.2#1	A cable plug capable of SOP' Commnications shall only detect and communicate with packets starting with SOP'	CAB_PROT_DISCOV_1 CAB_PROT_DISCOV_3 CAB_PROT_DISCOV_5 CAB_PROT_DISCOV_7	
5.6.1.2.2#2	The DFP or Source shall coordinate SOP and SOP' communication as to avoid collisions.		
5.6.1.2.2#3	For a cable plug supporting SOP’ Communications, if a valid SOP’ is not detected then the whole transmission shall be Discarded.	CBL_PHY_MSG	
5.6.1.2.2#4	For the DFP or Source supporting SOP’ Communications if a valid SOP or SOP’ is not detected then the whole transmission shall be Discarded.	BMC_PHY_MSG	

R2.0 Assertion #	Description	Test	Comments
5.6.1.2.2#5	When there is an Explicit Contract in place a UFP shall not send SOP' Packets and shall discard all packets starting with SOP'.	BMC_PHY_MSG_2 BMC_PHY_MSG_4	
5.6.1.2.2#6	When there is no Explicit Contract or an Implicit Contract in place a Sink shall not send SOP' Packets and shall discard all packets starting with SOP'.	BMC_PHY_MSG_2 BMC_PHY_MSG_4	
5.6.1.2.3 Start of Packet Sequence Prime (SOP'')			
5.6.1.2.3#1	A cable plug capable of SOP'' Communication, shall have a SOP' Communication capability in the other Cable plug.		
5.6.1.2.3#2	No cable shall only support SOP'' Communication.	CAB_PROT_DISCOV	
5.6.1.2.3#3	A cable plug to which SOP'' Communication is assigned shall only detect and communicate with packets starting with SOP'' and shall discard any other packets.	CAB_PHY_MSG	
5.6.1.2.3#4	A DFP which supports SOP'' Communication shall also support SOP' Communication and shall coordinate SOP* Communication so as to avoid collisions.	BMC_PHY_MSG	
5.6.1.2.3#5	For the cable plug supporting SOP'' Communication, if a valid SOP'' is not detected then the whole transmission shall be Discarded.	CAB_PHY_MSG	
5.6.1.2.3#6	For the DFP if a valid SOP* is not detected then the whole transmission shall be Discarded.	BMC_PHY_MSG	
5.6.1.2.3#7	A UFP shall not send SOP'' Packets and shall discard all Packets starting with SOP''.	BMC_PHY_MSG_3 BMC_PHY_MSG_5	
5.6.1.3 Packet Payload			
5.6.1.3#1	The packet payload...shall be encoded with the hex data codes from Table 5_1.	BMC_PHY_MSG_11	
5.6.1.4 CRC			
5.6.1.4#1	The CRC shall be inserted just after the payload.		
5.6.1.5 End of Packet (EOP)			
5.6.1.5#1	The end of packet marker shall be a single EOP K_code as defined inTable 5_1.	CAB_PHY_MSG BMC_PHY_MSG	
5.6.1.5#2	This shall mark the end of the CRC.	CAB_PHY_MSG BMC_PHY_MSG	
5.6.1.5#3	After the EOP, the CRC_residual shall be checked.	CAB_PHY_MSG BMC_PHY_MSG	
5.6.1.5#4	If the CRC is not good, the whole transmission shall be discarded	BMC_PHY_MSG_7 BMC_PHY_MSG_8	
5.6.1.5#5	if it is good, the packet shall be delivered to the Protocol Layer.	BMC_PHY_MSG_1	
5.6.2 CRC			
5.6.2#1	The Message header and data shall be protected by a 32_bit CRC.		
5.6.2#2	The CRC_32 polynomial shall be = 04C1 1DB7h.		
5.6.2#3	The CRC_32 Initial value shall be = FFFF FFFFh.		
5.6.2#4	CRC_32 shall be calculated for all bytes of the payload not inclusive of any packet framing symbols (i.e. excludes the preamble, SOP*, EOP).		
5.6.2#5	CRC_32 calculation shall begin at byte 0 bit 0 and continue to bit 7 of each of the bytes of the packet.		
5.6.2#6	The remainder of CRC_32 shall be complemented.		
5.6.2#7	The residual of CRC_32 shall be C704 DD7Bh.		
5.6.2#8	The output bit ordering shall be as detailed in Table 5_10		
5.6.2#9	The CRC_32 shall be encoded before transmission.		
5.6.3 Packet Detection Errors			
5.6.3#1	CRC errors, or errors detected while decoding encoded symbols using the code table, shall be treated the same way; the message shall be discarded and a GoodCRC message shall not be returned.	BMC_PHY_MSG_7 BMC_PHY_MSG_8 BMC_PHY_MSG_9 BMC_PHY_MSG_10	
5.6.3#2	While the receiver is processing a packet, if at any time VBUS becomes idle the receiver shall stop processing the packet and discard it (no GoodCRC message is returned).		

R2.0 Assertion #	Description	Test	Comments
5.6.4 Hard Reset			
5.6.4#1	A Device shall perform a Hard Reset when it receives Hard Reset signaling.	PROT_PROC_HR_TSTR_6	
5.6.4#2	After receiving the Hard Reset signaling, the device shall reset as described in the Protocol Layer, see Section 6.7.2.	PROT_PROC_HR_TSTR	
5.6.4#3	If a valid Hard Reset is not detected (see Table 5_3) then the whole transmission shall be ignored.	CAB_PHY_MSG BMC_PHY_MSG	
5.6.4#4	A Cable Plug shall perform a Hard Reset when it detects Hard Reset signaling being sent between the Port Partners.	CAB_PHY_MSG_14 CAB_PHY_MSG_16	
5.6.4#5	After receiving the Hard Reset signaling, the device shall reset as described in Section 6.7.2	CAB_PHY_MSG BMC_PHY_MSG	
5.6.4#6	The procedure for sending Hard Reset signaling shall be as follows: 1. If the PHY Layer is currently sending a message, the message shall be interrupted by sending an EOP K_code and the rest of the message discarded. 2. If VBUS is not idle, wait for it to become idle (see Section 5.8.2.6.4 for the definition of BFSK idle and Section 5.8.3.6.1. for the definition of BMC idle) 3. Wait tInterFrameGap. 4. If Vbus is still idle send the Preamble followed by the 4 K_codes for Hard Reset signaling. 5. Disable the channel (i.e. stop sending and receiving), reset the PHY Layer and inform the Protocol Layer that the PHY Layer has been reset. 6. Re_enable the channel when requested by the Protocol Layer.		
5.6.5 Cable Reset			
5.6.5#1	Cable Reset shall only be sent by the DFP.		
5.6.5#2	The state of the Cable Plug after the Cable Reset signaling shall be equivalent to power cycling the Cable Plug.	PROT_MSG_HDR_3 CAB_PHY_MSG_14	
5.7 Collision Avoidance			
5.7#1	The PHY Layer shall monitor the channel for data transmission and only initiate transmissions when VBUS or CC is idle.	BMC_PHY_RX_BUSIDL_2	
5.7#2	If the bus idle condition is present, it shall be considered safe to start a transmission provided the conditions detailed in Section 5.8.1.4 are met.	BMC_PHY_RX_BUSIDL_3	
5.7#3	The bus idle condition shall be checked immediately prior to transmission.	BMC_PHY_RX_BUSIDL	
5.7#4	If transmission cannot be initiated then the packet shall be discarded.	Vendor declaration	
5.7#5	If the packet is discarded because VBUS or CC is not idle, the PHY Layer shall signal to the protocol layer that it has discarded the message as soon VBUS or CC becomes idle.	not tested	
5.8.1.1 Common Signalling Parameters			
5.8.1.1#1	The electrical requirements specified in Table 5_13 shall apply to both the transmitter and receiver.	PHYFSK_TX_FREQ	
5.8.1.1#1a	Table 5_13: fBitRate		
5.8.1.2 Common Transmitter Signailing Specifications			
5.8.1.2#1	Time before the start of the first bit of the Preamble when the transmitter shall start driving the line.		
5.8.1.2.1 Bit Rate Drift			
5.8.1.2.1#1	The change in fBitRate during a packet shall be less than pBitRate.	BMC_PHY_TX_BIT_4 CAB_PHY_TX_BIT_4	
5.8.1.2.1#2	fBitRate throughout the packet, including the EOP, shall be within pBitRate of refBitRate.	BMC_PHY_TX_BIT_3 CAB_PHY_TX_BIT_3	
5.8.1.2.1#3	The transmitter shall have the same pBitRate for all packet types.	not tested	
5.8.1.4 Inter_Frame Gap			

R2.0 Assertion #	Description	Test	Comments
5.8.1.4#1	The transmitter shall drive the bus for no longer than tEndDriveBFSK or tEndDriveBMC (as appropriate) after transmitting the final bit of the Frame.		
5.8.1.4#2	Before starting to transmit the next Frame's Preamble the transmitter of the next Frame shall ensure that it waits for tInterFrameGap after either: 1. Transmitting the previous frame, or		
5.8.1.4#3	Before starting to transmit the next Frame's Preamble the transmitter of the next Frame shall ensure that it waits for tInterFrameGap after either: 2. Receiving the previous frame, or		
5.8.1.4#4	Before starting to transmit the next Frame's Preamble the transmitter of the next Frame shall ensure that it waits for tInterFrameGap after either: 3. Observing an idle condition on VBUS or CC (see Section 5.7).	CAB_PHY_RX_BUSIDL_2	
5.8.2 Binary Frequency Shift Keyed (BFSK) Signaling			
5.8.2#1	Continuous Phase Binary Frequency Shift Keying (BFSK) shall be used to encode bits for transmission on the channel.	PHYFSK_RX_BER	
5.8.2#2	In this specification, BFSK shall be understood to mean continuous phase BFSK.	not tested	
5.8.2#3	A signal of amplitude vTX shall be injected onto VBUS using a carrier frequency, fCarrier.	PHYFSK_TX_LEV PHYFSK_TX_FREQ	
5.8.2#4	The following logic states shall be used: _Logic 0 is indicated by a frequency fCarrier _ fDeviation. _Logic 1 is indicated by a frequency fCarrier + fDeviation.	PHYFSK_TX_FREQ	
5.8.2.1 Channel Overview			
5.8.2.1#1	The PHY Layer shall be AC coupled to VBUS or be tolerant of the maximum possible DC voltage that may be present on VBUS.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
5.8.2.1#2	While transmitting, the Source, or Sink, shall apply an AC signal with amplitude vTX to VBUS as measured between zIsolation and cAC_Coupling (VBUS).	PHYFSK_TX_LEV	
5.8.2.2 Transceiver isolation Impedance			
5.8.2.2#1	The Source and Sink shall place an isolation impedance between the VBUS wire bulk capacitance and the VBUS pin on the connector	PHYFSK_TERM	
5.8.2.2#2	The isolation impedance shall have an impedance of zIsolation at any frequency within fRange.	PHYFSK_TERM	
5.8.2.3 Transceiver AC Coupling Capacitance			
5.8.2.3#1	The Source and Sink shall be AC coupled to VBUS or be tolerant of the maximum possible DC voltage that may be present on VBUS.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
5.8.2.4.1 BFSK Common Parameters			
5.8.2.4.1#1	The electrical requirements specified in Table 5_16 shall apply to both the transmitter and receiver.	PHYFSK_TX_FREQ	
5.8.2.4.1#1a	Table 5_16: fCarrier		
5.8.2.4.1#1b	Table 5_16: fDeviation		
5.8.2.5 BFSK Transmitter Specification			
5.8.2.5#1	The transceiver shall terminate the VBUS line with rTX while it is powered, whether it is transmitting, receiving or waiting for the squelch to close.		
5.8.2.5.1 BFSK Transmitter Requirements			
5.8.2.5.1#1	The requirements specified in Table 5 18 shall apply to the transmitter.		
5.8.2.5.1#1a	Table 5_18: pCarrierFreq		

R2.0 Assertion #	Description	Test	Comments
5.8.2.5.1#1b	Table 5_18: pDevFreq		
5.8.2.5.1#1c	Table 5_18: rTX		
5.8.2.5.1#1d	Table 5_18: vTX		
5.8.2.5.1.2 Carrier_Frequency Drift			
5.8.2.5.1.2#1	The change in fCarrier during a packet shall be less than pCarrierFreq	PHYFSK_TX_FREQ	
5.8.2.5.1.2#2	fCarrier during the packet, including the EOP, shall be within pCarrierFreq of the reference carrier frequency.	PHYFSK_TX_FREQ	
5.8.2.5.1.2#3	The transmitter shall have the same pCarrierFreq for all packet types.	not tested	
5.8.2.5.1.3 Deviation_Frequency Drift			
5.8.2.5.1.3#1	The change in fDeviation shall be less than pDevFreq	PHYFSK_TX_FREQ	
5.8.2.5.1.3#2	fDeviation during the packet, including the EOP, shall be within pDevFreq of the reference deviation frequency.	PHYFSK_TX_FREQ	
5.8.2.5.1.3#3	The transmitter shall have the same fDeviation for all packet types.	not tested	
5.8.2.5.2 Transmitter Characteristics			
5.8.2.5.2#1	The transmitted waveform shall fulfill the eye diagram mask in Figure 5_11.	BMC_PHY_TX_EYE_3	
5.8.2.5.2#2	When starting to transmit a frame the transmitter shall enable its carrier tStartDrive before the start of the first preamble bit.		
5.8.2.5.2#3	It shall stop transmitting a carrier within tEndDriveBFSK of the end of the last transmitted symbol.		
5.8.2.5.2#4	the emitted spectrum, on VBUS at the transmitting device receptacle, shall comply with the mask in Figure 5 12 when VBUS is terminated by a nominal rTX at the connector through a cable no longer than 250mm whose characteristic impedance matches the termination impedance.	PHYFSK_TX_SPECT	
5.8.2.5.2#5	Normal rules and regulations for noise emissions shall still be applicable.	not tested	
5.8.2.5.2#6	Side lobes outside the coverage of Figure 5_12 shall be kept below the level as the figure shows.	PHYFSK_TX_SPECT	
5.8.2.6.1 BFSK Receiver Electrical Parameters			
5.8.2.6.1#1	The requirements specified in Table 5_20 shall apply to the receiver (except vSquelchDetecting).	PHYFSK_RX_SQU	
5.8.2.6.1#2	In the Squelch Operating mode the receiver shall meet the requirement nBER when the signal level exceeds vSquelchOperating.	PHYFSK_RX_SQU PHYFSK_RX_BER	
5.8.2.6.1#3	The receiver shall meet the nBER performance requirement in Table 5 20 when the voltage received on VBUS is within the allowable range of vRX.	PHYFSK_RX_BER	
5.8.2.6.1#4	Table 5_20, tBitStreamComplete		
5.8.2.6.2 Receiver Filter Specification			
5.8.2.6.2#1	The design of the receiver filter represented by the "Filter Bandpass" in Figure 5 9 is implementation specific, but shall take into account the out_of_band power_supply noise (see Sections 7.1.13.2 and 7.2.9.2).	PHYFSK_RX_BER	
5.8.2.6.3 Crosstalk in the cables			
5.8.2.6.3#1	In order to maintain good communications, the cables shall fulfill the crosstalk requirements in Section 3.6.6.	Cabcon question	
5.8.2.6.4 Definition of Idle			
5.8.2.6.4#1	For BFSK Signaling Scheme VBUS shall be declared idle when the signal level is less than vSquelchOperating.		
5.8.2.6.4#2	The power supply noise allowed by Figure 7_8 and Figure 7_11 shall not cause the receiver to indicate the channel is busy.	PHYFSK_RX_BER BMC_PHY_RX_INT_REJ_2 BMC_PHY_RX_INT_REJ_3	
5.8.2.7 Bit Stream			
5.8.2.7#1	The transmitter of the Provider/Consumer that implements		

R2.0 Assertion #	Description	Test	Comments
	Dead Battery Support shall be able to transmit a Bit Stream consisting of alternating "0s" and "1s" which may be viewed as concatenating multiple preambles as shown in Figure 5 13 (note that the last preamble may not contain 64 bits)		
5.8.2.7#2	The PHY shall continue to transmit the Bit Stream until the PD System is ready for a Source Capabilities message or VBUS becomes vSafe0V (see Figure 4 1).		
5.8.2.7#3	The Consumer/Provider's receiver shall declare a Bit Stream is detected after detecting 128 consecutive bits that match the preamble pattern of alternating "0s" and "1s".		
5.8.2.7#4	After the Bit Stream is detected, the receiver shall indicate that the Bit Stream has stopped when the squelch has closed (the signal level is below vSquelchOperating) for tBitStreamComplete.		
5.8.3.1 Encoding and signaling			
5.8.3.1#1	The USB PD baseband signal shall be driven on the CC wire with a tri_state driver that shall cause a vSwing swing on CC.	CAB_PHY_TERM	
5.8.3.1#2	When sending the preamble the transmitter shall start by transmitting a low level.	PHY_MSG_GEN	
5.8.3.1#3	The receiver shall tolerate the loss of the first edge.		
5.8.3.1#4	The transmitter shall terminate the final bit of the Frame by an edge (the "trailing edge") to help ensure that the receiver clocks the final bit.		
5.8.3.1#5	If the trailing edge results in the transmitter driving CC low (i.e. the final UI of the EOP is high), then the transmitter: 1. Shall continue to drive CC low for tHoldLowBMC. 2. Then shall continue to drive CC low for tEndDriveBMC measured from the trailing edge of the final bit of the Frame. 3. Then shall release CC to high impedance.		
5.8.3.1#6	If the trailing edge results in the transmitter driving CC high (i.e. the final half_UI of the frame is low), then the transmitter: 1. Shall continue to drive CC high for 1 UI. 2. Then shall drive CC low for tHoldLowBMC. 3. Then shall continue to drive CC low for tEndDriveBMC measured from the final edge of the final bit of the Frame. 4. Then shall release CC to high impedance.		
5.8.3.2.1 Transmit Masks			
5.8.3.2.1#1	The transmitted signal shall not violate the masks defined in Figure 5_22 and Figure 5_23, Table 5_21 and Table 5_22 at the output of a load equivalent to the cable model and receiver load model described in Section 5.8.3.3	CAB_PHY_TX_EYE_3	
5.8.3.2.1#2	The transmitted signal shall have a rise time no faster than tRise.	CAB_PHY_TX_EYE BMC_PHY_TX_EYE	
5.8.3.2.1#3	The transmitted signal shall have a fall time no faster than tFall.	CAB_PHY_TX_EYE BMC_PHY_TX_EYE	
5.8.3.2.2 Receive Masks			
5.8.3.2.2#1	A Provider using the BMC Signaling Scheme shall be capable of receiving a signal that complies with the mask when sourcing power as defined in Figure 5-25, Figure 5-26 and Table 5-23.	CAB_PHY_TX_EYE BMC_PHY_TX_EYE	
5.8.3.2.2#2	A Consumer using the BMC Signaling Scheme shall be capable of receiving a signal that complies with the mask when sinking power as defined in Figure 5-29, Figure 5-30 and Table 5-23.	CAB_PHY_TX_EYE BMC_PHY_TX_EYE	
5.8.3.2.2#3	Every product using the BMC Signaling Scheme shall be capable of receiving a signal that complies with the mask when power neutral as defined in Figure 5-27, Figure 5-28 and Table 5-23.	CAB_PHY_TX_EYE BMC_PHY_TX_EYE	
5.8.3.2.2#4	Dual-Role Devices shall meet the receiver requirements for a Provider when providing power during any transmission using the BMC Signaling Scheme or a Consumer when consuming power during any transmission using the BMC Signaling Scheme.	CAB_PHY_TX_EYE BMC_PHY_TX_EYE	

R2.0 Assertion #	Description	Test	Comments
5.8.3.2.2#5	Cable Plugs shall meet the receiver requirements for both a Provider and a Consumer during any transmission using the BMC Signaling Scheme.	CAB_PHY_TX_EYE	
5.8.3.2.2#6	The receiver sensitivity shall be set such that the receiver does not treat noise on an undriven signal path as an incoming signal.	CAB_PHY_RX_INT_REJ BMC_PHY_RX_INT_REJ	
5.8.3.2.2#7	Signal amplitudes below vNoiseIdle max shall be treated as noise when BMC is idle.	CAB_PHY_RX_BUSIDL BMC_PHY_RX_BUSIDL	
5.8.3.3 Transmitter Load Model			
5.8.3.3#1	The transmitter load model for the eye mask tests and vSwing shall be equivalent to the circuit outlined in Figure 5-31 for a Source and Figure 5-32 for a Sink.		
5.8.3.3#2	The value of the modeled cable inductance, La, (in nH) shall be calculated from the following formula: La = tCableDelaymax * zCablemin		
5.8.3.3#3	The value of the modeled cable capacitance, Ca, (in pF) shall be calculated from the following formula: Ca = (tCableDelaymax)/(zCablemin)		
5.8.3.3#4	cCablePlug models the capacitance of the plug at each end of the cable. cReceiver models the capacitance of the receiver. The maximum values shall be used in each case.		
5.8.3.4.1 BMC Common Parameters			
5.8.3.4.1#1	The electrical requirements specified in Table 5 24 shall apply to both the transmitter and receiver.		
5.8.3.5BMC Transmitter Specifications			
5.8.3.5#1	The transmitter shall meet the specifications defined in Table 5 25.		
5.8.3.5#2	cReceiver is the capacitance that a DFP or UFP shall present on the CC line when the DFP or UFP’s receiver is not transmitting on the line.		
5.8.3.5#3	Once transmission is complete, the transmitter shall disengage capacitance in excess of cReceiver from the CC wire within tInterFrameGap.		
5.8.3.5#4	cShunt shall not cause a violation of cReceiver when not transmitting.		
5.8.3.6 BMC Receiver Specifications			
5.8.3.6#1	The transmitter shall meet the specifications defined in Table 5 26.		
5.8.3.6#2	The DFP or UFP system shall have capacitance within this range (cReceiver) when not transmitting on the line.		
5.8.3.6.1 Definition of Idle			
5.8.3.6.1#1	After waiting tTransitionWindow without detecting nTransitionCount transitions the bus shall be declared idle.		
5.8.3.6.2 Multi_Drop			
5.8.3.6.2#1	The Multi_Drop transceiver shall obey all the electrical characteristics specified in this section except for those relating to capacitance.		
5.8.3.6.2#2	the maximum capacitance on the CC line contained within the UFP shall still be within cReceiver except when transmitting.		
5.8.4 Interoperability with BFSK and BMC			
5.8.4#1	Products with Type_C connectors shall not support BFSK without supporting BMC.		
5.8.4#2	A Source shall first attempt to become Connected with its Port Partner, using BMC; the attempt failing when the Source enters the PE_SRC_Disabled state for a Source (see Section 8.3.3.2).		
5.8.4#3	A Sink shall be able to receive, in the PE_SNK_Wait_for_Capabilities state (see Section 8.3.3.3), a Source Capabilities message sent either over the CC wire using BMC or over VBUS using BFSK.		

R2.0 Assertion #	Description	Test	Comments
5.8.4#4	The Sink, after the SinkWaitCapTimer has timed out in the PE_SNK_Wait_for_Capabilities state (see Section 8.3.3.3), shall issue Hard Reset signaling over both BMC and BFSK.		
5.8.4#5	Once the Port Partners are Connected they shall continue to use the same signaling scheme ,either BMC or BFSK, until a Detach or Hard Reset occurs.		
5.8.4#6	If either Port Partner issues Hard Reset signaling it shall issue Hard Reset signaling over both BMC and BFSK.		
5.9.1 BIST PRBS Pattern			
5.9.1#1	The generator polynomial for the PRBS_8 pattern shall be x8 + x6 + x5 + x4 + 1.		
5.9.1#2	The preloaded pattern shall be "all ones" i.e. all 8 bits in the shift register shall be set to "1".		
5.9.1#3	The pattern shall be preloaded when the request to enter test mode is given or received.		
5.9.1#4	At least nBISTConfidence of these Teat Frames shall be sent/received without error (see Section 5.9.1.1).		
5.9.1#5	The PRBS data shall be continued without change in the PRBS generator between Test Frames.		
5.9.1#6	If the payloads from all test frames were concatenated the resulting stream shall look like it was generated directly by the BIST generator.		
5.9.1#7	The Test Frame shall have a fixed length		
5.9.1#8	the only other signaling that shall be recognized in the test mode is the Hard Reset signaling, which shall be used to exit the test mode.		
5.9.1.1 Test Pattern Frame Transmission			
5.9.1.1#1	To reach this level of confidence, a minimum of nBISTConfidence Test Frames shall be transmitted.	not tested	
5.9.1.1#2	To end the test sequence, Hard Reset signaling shall be sent.		
5.9.1.1#3	If errors are detected more bits shall be sent , see [Maxim37]).	not tested	
5.9.1.2 Error Counters			
5.9.1.2#1	The UUT shall maintain a count of errors detected BISTErrorCounter (See Section 6.6.5).		
5.9.1.2#2	The number of errors shall be compared to the number of errors expected from the number of sent bits and the allowed error rate.	not tested	
5.9.2 BIST Carrier Mode 0			
5.9.2#1	In BIST Carrier Mode 0 the Physical Layer shall send out a continuous string of "0"s.		
5.9.3 BIST Carrier Mode 1			
5.9.3#1	In BIST Carrier Mode 1 the Physical Layer shall send out a continuous string of "1"s.		
5.9.4 BIST Carrier Mode 2			
5.9.4#1	In BIST Carrier Mode 2 the Physical Layer shall send out a continuous string of alternating "1"s and “0”s.	BMC_PHY_TX_BIT_2 BMC_PHY_TX_EYE_2 CAB_PHY_TX_EYE_2 CAB_PHY_TX_BIT_2	
5.9.5 BIST Carrier Mode 3			
5.9.5#1	In BIST Carrier Mode 3, the Physical Layer shall send out a continuous string of sixteen "1"s, followed by sixteen “0”s, followed by sixteen “1”s, etc.		
5.9.6 BIST Eye Pattern			
5.9.6#1	In BIST Eye Pattern the Physical Layer shall send out a continuous string of bits in accordance with section 5.9.1.		
5.9.7 BIST Test Data			
5.9.7#1	Figure 5_38 shows the Test Data Frame which shall be sent by the Tester to the UUT.	CAB_PHY_TERM BMC_PHY_TERM	
5.9.9 BIST Test Applicability			

R2.0 Assertion #	Description	Test	Comments
5.9.9#1	Table 5_29 shows the BIST Modes which shall be supported, depending on the signaling scheme (BMC or BFSK) supported by a device.		

7.4.2 Recommended (Should)

R2.0 Assertion #	Description	Test	Comments
SECTION 5 Physical Layer			
Subsection 5.2 Physical Layer Functions			
5.2#r1	In the active mode, where any of the functions listed above may be executed, the Phy Layer Block power consumption should be minimized.		
5.2#r2	In the squelch mode, when only the squelch detector is required, the power consumption should be minimized.		
5.8.2.5.1.1 Carrier Noise			
5.8.2.5.1.1#r1	The carrier phase noise should be considered when designing the transmitter system in order to achieve the required BER (nBER).		

7.5 Chapter 6 Protocol Layer

7.5.1 Normative (Shall/Normative)

R2.0 Assertion #	Description	Test	Comments
SECTION 6 Protocol Layer			
Subsection 6.2 Messages			
Subsection 6.2.1 Message Construction			
6.2.1#1	All messages shall be composed of a message header and a variable length (including zero) data portion.	PROT_HDR PROT_HDR_GCRC	
6.2.1.1 Message Header			
6.2.1.1#1	Every message shall start with a 16_bit message header, as defined in Figure 6_1.	PROT_HDR PROT_HDR_GCRC	
6.2.1.1#2	(Message Header Bits) Shall be set to 0	PROT_MSG_HDR_1 PROT_MSG_HDR_16 PROT_MSG_HDR_GCRC_1 PROT_MSG_HDR_GCRC_12	
6.2.1.2 Number of Data Objects			
6.2.1.2#1	The 3_bit Number of Data Objects field shall indicate the number of 32_bit Data Objects that follow the message header.	PROT_MSG_HDR_2 PROT_MSG_HDR_GCRC_2 PROT_MSG_CTRL_1 PROT_MSG_DATA_REQ_1 PROT_MSG_DATA_VDM_ID_ACK_1 PROT_MSG_DATA_VDM_SVID_INIT_1 PROT_MSG_DATA_VDM_SVID_ACK_1 PROT_MSG_DATA_VDM_MODE_INIT_1 PROT_MSG_DATA_VDM_MODE_ACK_1 PROT_MSG_DATA_VDM_ENTER_MODE_1 PROT_MSG_DATA_VDM_EXIT_MODE_1 PROT_MSG_DATA_VDM_ATT_1 PROT_MSG_DATA_SRC_CAP_1 PROT_MSG_DATA_BIST_1	
6.2.1.2#2	The Number of Data Objects field shall apply to all SOP* Packet types.		
6.2.1.3 MessageID			
6.2.1.3#1	The MessageIDCounter shall be initialized to zero at power_on, as a result of a Soft Reset or a Hard Reset.	PROT_MSG_HDR_3 PROT_MSG_HDR_4 PROT_MSG_HDR_5 PROT_MSG_HDR_GCRC_3 BMC_PHY_MSG_14	
6.2.1.3#2	The MessageIDCounter shall be incremented when a message is successfully received as indicated by receipt of a GoodCRC message.	PROT_MSG_HDR_6 BMC_PHY_MSG_13	
6.2.1.3#3	The MessageID field shall apply to all SOP* Packet types.		
6.2.1.4 Port Power Role			
6.2.1.4#1	The 1_bit Port Power Role field shall indicate the current power role of the port: • 0b Sink • 1b Source	PROT_MSG_HDR_11 PROT_MSG_HDR_GCRC_7	
6.2.1.4#2	Messages, such as Ping, and GotoMin, that are only ever sent by a Source, shall always have the Port Power Role field set to Source.	PROT_MSG_HDR_11 PROT_PROC_PING_1	
6.2.1.4#3	Similarly messages such as Request that are only ever sent by a Sink, shall always have the Port Power Role field set to Sink.	PROT_MSG_HDR_11	
6.2.1.4#4	During the Power Role Swap Sequence, for the initial Source Port, the Port Power Role field shall be set to Sink in the PS_RDY message indicating that the initial Source's power supply is turned off (see Figure 8_6 and Figure 8_7).	PROT_MSG_HDR_11	
6.2.1.4#5	During the Power Role Swap Sequence, for the initial Sink Port, the Port Power Role field shall be set to Source for messages initiated by the Policy Engine after receiving the PS_RDY message from the initial Source (see Figure 8_6 and Figure 8_7).	PROT_MSG_HDR_11	

R2.0 Assertion #	Description	Test	Comments
6.2.1.4#6	The Port Power Role field of a received Message shall not be verified by the receiver and no error recovery action shall be taken if it is incorrect.		
6.2.1.4#7	The Port Power Role field shall only be defined for SOP Packets.	PROT_MSG_HDR_12 PROT_MSG_HDR_GCRC_8	
6.2.1.5 Specification Revision			
6.2.1.5#1	The 2_bit Specification Revision field shall indicate the revision of the Power Deliver Specification supported by the Device. • 00b – Revision 1.0. • 01b _ Revision 2.0. •10b _ 11b – Reserved.	PROT_MSG_HDR_13 PROT_MSG_HDR_18 PROT_MSG_HDR_GCRC_9 PROT_MSG_HDR_GCRC_14	
6.2.1.5#2	On receipt of a message header with a higher revision number than that supported, a port shall respond using the highest revision number it supports.	PROT_REV_NUM	
6.2.1.5#3	The Specification Revision field shall apply to all SOP* Packet types.		
6.2.1.6 Port Data Role			
6.2.1.6#1	The 1_bit Port Data Role field shall indicate the current data role of the Port: · 0b UFP · 1b DFP	PROT_MSG_HDR_14 PROT_MSG_HDR_GCRC_10	
6.2.1.6#2	The Port Data Role field shall only be defined for SOP Packets.	PROT_MSG_HDR_15	
6.2.1.6#3	For all other SOP* Packets the Port Data Role field is Reserved and shall be set to zero.	PROT_MSG_HDR_GCRC_11	
6.2.1.6#4	Should a Type-C Port receive a Message with the Port Data Role field set to the same Data Role as its current Data Role, except for the GoodCRC Message, Type-C error recovery actions as defined in [USBType-C 1.0] shall be performed.		
6.2.1.6#5	For a Type-C Port the Port Data Role field shall be set to the default value at attachment after a Hard Reset: 0b for a Port with Rd asserted and 1b for a Port with Rp asserted.		
6.2.1.7 Cable Plug			
6.2.1.7#1	The 1_bit Cable Plug field shall indicate whether this message originated from a Cable Plug: · 0b message originated from a DFP or UFP · 1b message originated from a Cable Plug	PROT_MSG_HDR_GCRC_8	
6.2.1.7#2	The Cable Plug field shall only apply to SOP’ and SOP’’ Packet types.	PROT_MSG_HDR_GCRC_8 CAB_PROT_DISCOV_1 CAB_PROT_DISCOV_3 CAB_PROT_DISCOV_5 CAB_PROT_DISCOV_7	
6.2.1.8 Message Type			
6.2.1.8#1	The 4_bit Message Type field shall indicate the type of message being sent.	PROT_MSG_HDR_17 PROT_MSG_DATA_REQ_2 PROT_MSG_DATA_VDM_ID_INIT_2 PROT_MSG_DATA_VDM_ID_ACK_2 PROT_MSG_DATA_VDM_SVID_INIT_2 PROT_MSG_DATA_VDM_SVID_ACK_2 PROT_MSG_DATA_VDM_MODE_INIT_2 PROT_MSG_DATA_VDM_MODE_ACK_2 PROT_MSG_DATA_VDM_ENTER_MODE_2 PROT_MSG_DATA_VDM_EXIT_MODE_2 PROT_MSG_DATA_VDM_ATT_2 PROT_MSG_DATA_SRC_CAP_2 PROT_MSG_DATA_BIST_2 PROT_MSG_DATA_VEND_1	
6.2.1.8#2	The Message Type field shall apply to all SOP* Packet types.		
6.3 Control Message			
6.3#1	Table 6_2: The Sent by column indicates entities which may send the given message (Source, Sink or Cable Plug); entities	PROT_MSG_HDR_17	

R2.0 Assertion #	Description	Test	Comments
	not listed shall not issue the corresponding message.		
6.3#2	Table 6_2: The Valid Signaling column indicates the messages which shall only be issued in SOP Packets and the messages which may be issued in SOP* Packets	PROT_MSG_HDR_17	
6.3#3	(Table 6_2) All values not explicitly defined are reserved and shall not be used.		
6.3#4	(Table 6_2) Value is reserved and shall not be used.		
6.3.1 GoodCRC Message			
6.3.1#1	The GoodCRC message shall be sent by the receiver to acknowledge that the previous message was correctly received (i.e. had a good CRC).	PROT_MSG_HDR_GCRC	
6.3.1#2	The GoodCRC message shall return the message's MessageID so the transmitter can determine that the correct message is being acknowledged.	PROT_MSG_HDR_GCRC_3 BMC_PHY_MSG_12 BMC_PHY_MSG_13	
6.3.1#3	The first bit of the GoodCRC message shall be returned within tTransmit after receipt of the last bit of the previous message.	PROT_MSG_HDR_GCRC_13 PROT_PROC_GOODCRC_UUT_1	
6.3.2 GotoMin Message			
6.3.2#1	The Sink shall respond to a GotoMin message by reducing its power consumption to the pre_negotiated value (Minimum Operating Current) within tSnkNewPower time.	PROT_PROC_GOTOMIN_TSTR	
6.3.2#2	The Source shall return power to the Sink(s) it has 'borrowed' from using the GotoMin mechanism before it can allocate any 'new' power to other devices.	A subject of the 'Test Jig for Multiple Port Testing'	
6.3.3 Accept Message			
6.3.3#1	It shall be sent by the Source to signal the Sink that the Source is willing to meet the Request message.	PROT_PROC_REQ_TSTR	
6.3.3#2	It shall be sent by the recipient of the PR Swap message to signal that it is willing to do a Power Role Swap and has begun the Power Role Swap sequence.	PROT_PROC_SWAP_TSTR_SRC_1 PROT_PROC_SWAP_TSTR_SNK_1	
6.3.3#3	It shall be sent by the recipient of the DR_Swap message to signal that it is willing to doa a Data Role Swap and has begun the Data Role Swap sequence.		
6.3.3#4	It shall be sent by the recipient of the VCONN_Swap message to signal that it is willing to doa a VCONN Swap and has begun the VCONN Swap sequence.		
6.3.3#5	It shall be sent by the recipient of the Soft Reset Message to indicate that it has completed its Soft Reset.	PROT_PROC_SR_TSTR_1	
6.3.3#6	The Accept message shall be sent within tReceiverResponse of the receipt of the last bit of the message (see section 6.5.3).	PROT_PROC_REQ_TSTR PROT_PROC_SWAP_TSTR_SNK_1 PROT_PROC_SWAP_TSTR_SRC_1	
6.3.4 Reject Message			
6.3.4#1	It shall be sent to signal the Sink that the Source is unable to meet the Request message.	BMC_POW_SRC_LOAD_P_PC_4	
6.3.4#2	It shall be sent by the recipient of a PR Swap message to indicate it is unable to do a Power Role Swap.	PROT_SEQ_SWAP_REJ	
6.3.4#3	It shall be sent by the recipient of a DR_Swap message to indicate it is unable to do a Data Role Swap.	BMC_PROT_SEQ_DRSWAP_1 BMC_PROT_SEQ_DRSWAP_3	
6.3.4#4	It shall be sent by the recipient of a VCONN_Swap message to indicate it is unable to do a VCONN Swap.	BMC_PROT_SEQ_VCSWAP_1 BMC_PROT_SEQ_VCSWAP_3	
6.3.4#5	It shall be sent by a Source without Dual_Role capability in response to a Get_Sink_Cap message.	BMC_PROT_SEQ_GETCAPS _3	
6.3.4#6	It shall be sent by a Sink without Dual_Role capability in response to a Get_Source_Cap message.	BMC_PROT_SEQ_GETCAPS_1	
6.3.4#7	The Reject message shall be sent within tReceiverResponse of the receipt of the last bit of the message (see Section 6.5.3).	PROT_SEQ_SWAP_REJ	
6.3.5.1 Pings on Type_A and Type_B Connectors			
6.3.5.1#1	Once a Power Delivery Contract is established Sources shall	PROT_PROC_PING_2	

R2.0 Assertion #	Description	Test	Comments
	periodically send the Ping message every tSourceActivity after the last message has been sent/received except when: • the system is not operating in USB Power Delivery mode (i.e. in standard [USB2.0], [USB3.1], [USBType_C1.0] or [BC1.2] operation). • A Provider or Provider/Consumer is operating as a Source at vSafe5V in the PE_SRC_Ready state (i.e. power negotiation has already taken place see Section 8.3.3.2.6).		
6.3.5.2 Pings on Type_C Connectors			
6.3.5.2#1	A Sink using a Type_C connector shall not expect to receive Ping messages but shall not treat Ping messages as an error if they are received.	PROT_MSG_CTRL_PING_2	
6.3.6 PS_RDY Message			
6.3.6#1	The PS_RDY message shall be sent by the Source (or by both the new Sink and new Source during the Power Role Swap sequence) to indicate its power supply has reached the desired operating condition.	PROT_PROC_SWAP_TSTR_SRC_3 PROT_PROC_SWAP_TSTR_SRC_4 PROT_PROC_SWAP_TSTR_SNK_5 PROT_PROC_SWAP_UUT_SRC_6 PROT_PROC_SWAP_UUT_SNK_5 PROT_PROC_SWAP_UUT_SNK_6	
6.3.7 Get_Source_Cap Message			
6.3.7#1	The port shall respond by returning a Source Capabilities message (see Section 6.4.1.1.1).	PROT_PROC_GETSRCCAPS_TST R	
6.3.7#2	A Sink port, without dual_role capability, shall return a Reject message.	PROT_SEQ_GETSRCCAP_REJ PROT_PROC_GETSRCCAPS_TST R	
6.3.8 Get_Sink_Cap Message			
6.3.8#1	The port shall respond by returning a Sink Capabilities message (see Section 6.4.1.1.2).	PROT_PROC_GETSNKCAPS_TST R	
6.3.8#2	A Source port, without dual_role capability, shall return a Reject message.	PROT_PROC_GETSNKCAPS_TST R	
6.3.9 DR_Swap Message			
6.3.9#1	[USBType_C1.0] DRPs shall have the capability to perform a Data Role Swap from the PE_SRC_Ready or PE_SNK_Ready states.		
6.3.9#2	A Data Role Swap shall be regarded in the same way as a cable detach/reattach in relation to any USB communication which is ongoing between the Port Partners and all active Modes shall be exited.		
6.3.9#3	If there are any Active Modes between the Port Partners when a DR_Swap Message is a received then a Hard Reset shall be performed (see Section 6.4.4.3.4).		
6.3.9#4	If the Cable Plug has any Active Modes then the DFP shall not issue a DR_Swap Message and shall cause all Active Modes in the Cable Plug to be exited before accepting a DR Swap request.		
6.3.9#5	The Source of VBUS and VCONN shall remain unchanged as well as the Rp/Rd resistors on the CC wire during the Data Role Swap process.		
6.3.9#6	The recipient of the Dr_Swap message shall respond by sending an Accept message, Reject message or Wait message.		
6.3.9#7	If an Accept message is sent, the Source and Sink shall exchange operational roles.		
6.3.9#8	If a Reject message is sent, the requester is informed that the recipient is unable, or unwilling, to do a Data Role swap and no action shall be taken.		
6.3.9#9	If a Wait message is sent, the requester is informed that a data role swap might be possible in the future but that no immediate action shall be taken.		

R2.0 Assertion #	Description	Test	Comments
6.3.9#10	Before a Data Role Swap the initial DFP shall have its Port Data Role bit set to DFP, and the initial UFP shall have its Port Data Role bit set to UFP		
6.3.9#11	After a successful Data Role Swap the DFP/Host shall become the UFP/Device and vice_versa;		
6.3.9#12	the new DFP shall have its Port Data Role bit set to DFP, and the new UFP shall have its Port Data Role bit set to UFP.		
6.3.10 PR Swap Message			
6.3.10#1	The recipient of the message shall respond by sending an Accept message, Reject message or Wait Message.	PROT_PROC_SWAP_TSTR_SRC_1 PROT_PROC_SWAP_TSTR_SNK_1	
6.3.10#2	If an Accept message is sent, the Source and Sink shall do a Power Role Swap.	PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_TSTR_SNK_2	
6.3.10#3	If a Reject message is sent, the requester is informed that the recipient is unable or unwilling to do a Power Role Swap and no action shall be taken.	Not tested	
6.3.10#4	If a Wait message is sent, the requester is informed that a Power Role Swap might be possible in the future but that no immediate action shall be taken.		
6.3.10#5	After a successful Power Role Swap the Port Partners shall reset their respective Protocol Layers (equivalent to a Soft Reset): resetting their MessageIDCounter, RetryCounter and Protocol Layer state machines before attempting to establish an Explicit Contract. At this point the Source shall also reset its CapsCounter.		
6.3.10#6	Since a UFP Source can attempt to send a Discover Identity Command using SOP’ to a Cable Plug prior to the establishment of an Explicit Contract, a DFP Sink shall disable the receiving of SOP’ Messages until an Explicit Contract has been established.		
6.3.10#7	For Type_C Ports the Source shall have Rp asserted on the CC wire and the Sink shall have Rd asserted on the CC wire.		
6.3.10#8	When performing a Power Role Swap from Source to Sink, a Type_C Port shall change is CC Wire resistor from Rp to Rd.		
6.3.10#9	When performing a Power Role Swap from Sink to Source, a Type_C Port shall change is CC Wire resistor from Rd to Rp.		
6.3.10#10	The DFP (Host), UFP (Device) roles and Source of VCONN shall remain unchanged during the Power Role Swap process.		
6.3.11 VCONN Swap Message			
6.3.11#1	The VCONN_Swap message shall only be sent by a DFP Sourcing VCONN to request an exchange of VCONN power sources.		
6.3.11#2	The VCONN_Swap Message may be sent by either Port Partner to request an exchange of VCONN Source.		
6.3.11#3	The recipient of the message shall respond by sending an Accept message, Reject message or Wait message.		
6.3.11#4	If an Accept message is sent, the Port Partners shall perform a VCONN Swap.		
6.3.11#5	The new VCONN source shall send a PS_RDY message within tVCONNOn to indicate that it now sourcing VCONN.		
6.3.11#6	The initial VCONN source shall cease sourcing VCONN within tVCONNOff of receipt of the last bit of the EOP of the PS_RDY message.		
6.3.11#7	If a Reject message is sent, the requester is informed that the recipient is unable, or unwilling, to do a VCONN Swap and no action shall be taken.		
6.3.11#8	If a Wait message is sent, the requester is informed that a VCONN Swap might be possible in the future but that no immediate action shall be taken.		

R2.0 Assertion #	Description	Test	Comments
6.3.11#9	The DFP (Host), UFP (Device) roles and Source of VBUS shall remain unchanged as well as the Rp/Rd resistors on the CC wire during the VCONN Swap process.		
6.3.11#10	Note: VCONN shall be continually sourced during the VCONN Swap process in order to maintain power to the Cable Plug(s) i.e. make before break.		
6.3.12 Wait Message			
6.3.12#1	It (Wait Message) shall be sent to signal the Sink that the Source is unable to meet the request at this time.		
6.3.12#2	It (Wait message) shall be sent by the recipient of a PR Swap message to indicate it is unable to do a Swap at this time		
6.3.12#3	It (Wait message) shall be sent by the recipient of a DR_Swap message to indicate it is unable to do a Data Role Swap at this time.		
6.3.12#4	It (Wait message) shall be sent by the recipient of a VCONN_Swap message to indicate it is unable to do a VCONN Swap at this time.		
6.3.12#5	It (Wait message) shall be sent within tReceiverResponse of the receipt of the last bit of the message (see Section 6.5.2).		
6.3.12.1 Wait in response to a Request Message			
6.3.12.1#1	A Source shall only send a Wait Message in response to a Request Message when an Explicit Contract exists between the Port Partners.		
6.3.12.2 Wait in response to a PR Swap Message			
6.3.12.2#1	In this case the hub shall send a Wait message in response to a PR Swap message from its Port Partner.		
6.3.13 Soft Reset Message			
6.3.13#1	The Soft Reset message shall cause a Soft Reset of the connected Port Pair (see section 6.7.1.		
6.3.13#2	If the Soft Reset message fails a Hard Reset shall be initiated within tHardReset of the last CRCReceiveTimer expiring after nRetryCount retries have been completed.		
6.3.13#3	If the error is not corrected by the Soft Reset, Hard Reset signaling shall be issued (see section 6.7).		
6.3.13#4	A Soft Reset message shall be targeted at a specific entity depending on the type of SOP* Packet used.	CAB_PHY_MSG BMC_PHY_MSG	
6.3.13#5	Soft Reset messages sent using SOP Packets shall Soft Reset the Port Partner only.	CAB_PHY_MSG	
6.3.13#6	Soft Reset messages sent using SOP'/SOP'' Packets shall Soft Reset the corresponding Cable Plug only.	CAB_PHY_MSG	
6.3.13#7	If the Source or DFP wants to communicate with a Cable Plug using SOP' Packets it shall issue a Soft Reset message using a SOP' Packet in order to reset the Cable Plug's Protocol Layer.		
6.3.13#8	If the Source or DFP wants to communicate with a Cable Plug using SOP'' Packets it shall issue a Soft Reset message using a SOP'' Packet in order to reset the Cable Plug's Protocol Layer.		
6.4 Data Message			
6.4#1	A Data Message shall consist of a message header and be followed by one or more Data Objects.	PROT_MSG_HDR	
6.4#2	The Sent by column indicates entities which may send the given message (Source, Sink or Cable Plug); entities not listed shall not issue the corresponding message.	PROT_PROC_GETSRCCAPS_TSTR_3 PROT_PROC_GETSNKCAPS_TSTR_3 BMC_PROT_SEQ_GETCAPS_1 BMC_PROT_SEQ_GETCAPS _3 BMC_PROT_SEQ_GETCAPS_1 BMC_PROT_SEQ_GETCAPS _3	
6.4#3	The Valid Signaling column indicates the messages which shall only be issued in SOP Packets and the messages which may be issued in SOP* Packets.		

R2.0 Assertion #	Description	Test	Comments
6.4#4	(Table 6_4) All values not explicitly defined are reserved and shall not be used.	PROT_HDR	
6.4.1 Capabilities Message			
6.4.1#1	A Capabilities message (Source Capabilities message or Sink Capabilities message) shall have at least one Power Data Object for vSafe5V.	PROT_MSG_DATA_SRC_CAP_10 PROT_PROC_SRCCAPS_UUT	
6.4.1#2	The Capabilities message shall also contain the sending port's information followed by up to 6 additional Power Data Objects.	PROT_MSG_DATA_SRC_CAP PROT_PROC_SRCCAPS_UUT	
6.4.1#3	Power Data Objects in a Capabilities message shall be sent in the following order: 1. The vSafe5V Fixed Supply Object shall always be the first object. 2. The remaining Fixed Supply Objects, if present, shall be sent in voltage order; lowest to highest. 3. The Battery Supply Objects, if present shall be sent in Minimum Voltage order; lowest to highest. 4. The Variable Supply (non battery) Objects, if present, shall be sent in Minimum Voltage order; lowest to highest.	PROT_MSG_DATA_SRC_CAP_14 PROT_PROC_SRCCAPS_UUT	
6.4.1#4	A list of one or more Power Data Objects shall be sent by the Source in order to convey its capabilities.		
6.4.1#5	Where Maximum an d Minimum Voltage and Current values are given in PDOs these shall be taken to be absolute.		
6.4.1#6	The Source and Sink shall not negotiate a power level that would allow the current to exceed the maximum current supported by their receptacles or the attached plug (see Section 3.1.1 and [USBType_C1.0]).		
6.4.1#7	The Source shall limit its offered capabilities to the maximum current supported by its receptacle and attached plug.		
6.4.1#8	A Sink with Type_A or a Type_B receptacle shall limit its requested capabilities to the maximum current supported by its receptacle and attached plug.		
6.4.1.1.1 Use by Sources			
6.4.1.1.1#1	Following a Hard Reset, a power_on event or plug insertion event, a Source port shall send a Source Capabilities message after every SourceCapabilityTimer timeout as an advertisement…		
6.4.1.1.1#2	Following a HardReset, a power_on event or plug insertion event...that shall be interpreted by the Sink port on attachment.		
6.4.1.1.1#3	The Source shall continue sending a minimum of nCapsCount Source Capabilities messages until a GoodCRC message is received.		
6.4.1.1.1#4	Additionally, a Source Capabilities message shall only be sent in the following cases: By the Source port from the PE_SRC_Ready state upon a change in its ability to supply power By a Source port or Dual-Role port in response to a Get_Source_Cap message.	PROT_MSG_DATA_SRC_CAP	
6.4.1.1.2 Use by Sinks			
6.4.1.1.2#1	A USB Power Delivery capable Sink, upon detecting vSafe5V on VBUS and after a SinkActivityTimer timeout without seeing a Capabilities message, shall send a HardReset.		
6.4.1.2 Source Capabilities Message			
6.4.1.2#1	A Source port shall report its capabilities in a series of 32_bit Power Data Objects (see Table 6_4) as part of a Source Capabilities message (see Figure 6_2).	PROT_MSG_DATA_SRC_CAP PROT_PROC_SRCCAPS_UUT	
6.4.1.2#2	Each Power Data Object shall describe a specific Source capability such as a battery (e.g. 2.8_4.1V) or a fixed power	PROT_MSG_DATA_SRC_CAP_13 PROT_PROC_SRCCAPS_UUT	

R2.0 Assertion #	Description	Test	Comments
	supply (e.g. 12V) at a maximum allowable current.		
6.4.1.2#3	The Number of Data Objects field in the header shall define the number of Power Data Objects that follow the Message Header in a Data Message.	PROT_MSG_DATA_SRC_CAP PROT_PROC_SRCCAPS_UUT	
6.4.1.2#4	All Sources shall minimally offer one Power Data Object that reports vSafe5V.	PROT_PROC_GETSRCCAPS_TSTR	
6.4.1.2#5	A Source shall not offer multiple Power Data Objects of the same type (fixed, variable, battery) and the same voltage but shall instead offer one Power Data Object with the highest available current for that Source capability and voltage.	BMC_POW_SRC_LOAD_P_PC_2 BMC_POW_SRC_LOAD_CP_ACC_3	
6.4.1.2#6	DRPs that support VCONN Powered Accessories do not source VBUS (see [USBType-C 1.0]) however when sourcing Vconn they shall advertise vSafe5V with the Maximum Current set to 0mA in the first Power Data Object.		
6.4.1.2#7	A Sink shall evaluate every Source Capabilities message it receives and shall respond with a Request message.	PROT_PROC_SRCCAPS_TSTR	
6.4.1.2#8	If its power consumption exceeds the Source's capabilities it shall re_negotiate so as not to exceed the Source's most recently advertised capabilities.		
6.4.1.2.1 Management of the Reserve			
6.4.1.2.1#1	Where a Power Reserve has been allocated to a Sink the Source shall indicate the Power Reserve as part of every Source Capabilities message it sends.		
6.4.1.2.1#2	When the same Power Reserve is shared between several Sinks the Source shall indicate the Power Reserve as part of every Source Capabilities message it sends to every Sink.		
6.4.1.2.1#3	When the Reserve is being temporarily used by a giveback capable Sink the Source shall shall indicate the Power Reserve as available in every Source Capabilities message it sends.		
6.4.1.2.1#4	However in this situation, when the Reserve is requested by a Sink, the Source shall return a Wait message while it retrieves this power using a GotoMin message.		
6.4.1.2.1#5	Once the additional power has been retrieved the Source shall send a new Source Capabilities message in order to trigger a new request from the Sink requesting the Power Reserve.		
6.4.1.2.1#6	The Power Reserve may be de_allocated by the Source at any time, but the de_allocation shall be indicated to the Sink or Sinks using the Power Reserve by sending a new Source Capabilities message.		
6.4.1.2.2 Receipt of Unexpected Capabilities Message			
6.4.1.2.2#1	When a Provider/Consumer port, with a Type-A receptacle, that is operating as a Source receives an unexpected Source Capabilities message (e.g. one not in response to a Get_Source_Cap message), it shall silently remove vSafe5V from VBUS within tPSSourceOff of receiving the Source Capabilities message EOP.	PROT_SRC_SRC	
6.4.1.2.3 Source Fixed Supply Power Data Object			
6.4.1.2.3#1	All other Fixed Supply Power Supply Data objects shall set bits 29...22 to zero.	PROT_MSG_DATA_SRC_CAP_15	
6.4.1.2.3#2	For a Source offering no capabilities, the Voltage (B19..10) shall be set to 5V and the Maximum Current shall be set to 0mA.	PROT_MSG_DATA_SRC_CAP_17	
6.4.1.2.3#3	When a Source wants a Sink, consuming power from VBUS, to go to its lowest power state, the Voltage (B19..10) shall be set to 5V and the Maximum Current shall be set to 0mA.	BMC_POW_SNK_TRANS_PC POW_SNK_TRANS_C_CP	
6.4.1.2.3#4	(Table 6_7 bits 24..22) Reserved – shall be set to zero.	PROT_MSG_DATA_SRC_CAP_9	
6.4.1.2.3.1 Dual_Role Power			

R2.0 Assertion #	Description	Test	Comments
6.4.1.2.3.1#1	The Dual_Role Power bit shall be set when the port is Dual_Role Power capable i.e supports the PR Swap message.	PROT_MSG_DATA_SRC_CAP_4	
6.4.1.2.3.1#2	This is a static capability which shall remain fixed for a given device.	PROT_MSG_DATA_SRC_CAP_4	
6.4.1.2.3.2 USB Suspend Supported			
6.4.1.2.3.2#1	Prior to a PD contract or when the USB Communications Capable bit is set to zero, this flag is undefined and Sinks shall follow the rules for suspend as defined in [USB2.0], [USB3.1], [USBType-C1.0] or [BC1.2].	USB2 and USB3 testing	
6.4.1.2.3.2#2	If the USB Suspend Supported flag is set, then the Sink shall follow the [USB2.0] or [USB3.1] rules for suspend and resume. A PDUSB peripheral device may draw up to pSnkSusppSnkSusp during suspend; a PDUSB Hub may draw up to pHubSusp during suspend (see Section 7.2.4).	PROT_MSG_DATA_SRC_CAP_5	
6.4.1.2.3.2#3	If the USB Suspend Supported flag is cleared, then the Sink shall ignore the [USB2.0] or [USB3.1] rules for suspend and may continue to draw the negotiated power. However, when USB is suspended, the USB device state is also suspended.		
6.4.1.2.3.3 Externally Powered			
6.4.1.2.3.3#1	The Externally Powered bit shall only be set when an AC "wall" supply is providing 100% of the source's power.	PROT_EXT_PWR PROT_MSG_DATA_SRC_CAP_6	
6.4.1.2.3.4 USB Communications Capable			
6.4.1.2.3.4#1	The USB Communications Capable bit shall only be set for devices capable of communication over the USB data lines (e.g. D+/_ or SS Tx/Rx).	PROT_MSG_DATA_SRC_CAP_7	
6.4.1.2.3.5 Data Role Swap			
6.4.1.2.3.5#1	The Data Role Swap bit shall be set when the Port is Type-C (see [USBType-C1.0]) and supports the DR_Swap message.	PROT_MSG_DATA_SRC_CAP_8	
6.4.1.2.3.5#2	This is a static capability which shall remain fixed for a given device.	PROT_MSG_DATA_SRC_CAP_8	
6.4.1.2.3.6 Peak Current			
6.4.1.2.3.6#1	This capability is intended for direct port to port connections only and shall not be offered to downstreams Sinks via a Hub.		
6.4.1.2.3.6#2	Every Fixed Supply PDO shall contain a Peak Current field.	PROT_MSG_DATA_SRC_CAP_10 PROT_MSG_DATA_SRC_CAP_16	
6.4.1.2.3.6#3	Supplies that want to offer a set of overload capabilities shall advertise this through the Peak Current field in the corresponding Fixed Supply PDO (see Table 6 7).	PROT_MSG_DATA_SRC_CAP_10	
6.4.1.2.3.6#4	Supplies that do not support an overload capability shall set these bits to 00b in the corresponding Fixed Supply PDO.	PROT_MSG_DATA_SRC_CAP_10	
6.4.1.2.4 Variable Supply (non battery) Power Data Object			
6.4.1.2.4#1	The voltage fields shall define the range that output voltage shall fall within.	PROT_MSG_DATA_SRC_CAP_19 PROT_MSG_DATA_SRC_CAP_20	
6.4.1.2.4#2	This does not indicate the voltage that will actually be supplied, except it shall fall within that range.	PROT_MSG_DATA_SRC_CAP_19 PROT_MSG_DATA_SRC_CAP_20	
6.4.1.2.4#3	The absolute voltage, including any voltage variation, shall not fall below the Minimum Voltage and shall not exceed the Maximum Voltage.		
6.4.1.2.5 Battery Supply Power Data Object			
6.4.1.2.5#1	The voltage fields shall represent the battery's voltage range.	PROT_MSG_DATA_SRC_CAP_22 PROT_MSG_DATA_SRC_CAP_23	
6.4.1.2.5#2	The battery shall be capable of supplying the Power value over the entire voltage range.	PROT_MSG_DATA_SRC_CAP_24	
6.4.1.2.5#3	The absolute voltage, including any voltage variation, shall not fall below the Minimum Voltage and shall not exceed the Maximum Voltage.		
6.4.1.3 Sink Capabilities Message			

R2.0 Assertion #	Description	Test	Comments
6.4.1.3#1	A Sink port shall report power levels it is able to operate at in a series of 32_bit Power Data Objects	PROT_MSG_DATA_SNK_CAP	
6.4.1.3#2	Each Power Data Object shall describe a specific Sink operational power level, such as a battery (e.g. 2.8_4.1V) or a fixed power supply (e.g. 12V).	PROT_MSG_DATA_SNK_CAP	
6.4.1.3#3	The Number of Data Objects field in the header shall define the number of Power Data Objects that follow the Message Header in a Data Message.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3#4	All Sinks shall minimally offer one Power Data Object with a power level at which the Sink can operate.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3#5	A Sink shall not offer multiple Power Data Objects of the same type (fixed, variable, battery) and the same voltage but shall instead offer one Power Data Object with the highest available current for that Sink capability and voltage.	PROT_MSG_DATA_SNK_CAP_13	
6.4.1.3#6	All Sinks shall include one Power Data Object that reports vSafe5V even if they require additional power to operate fully.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3#7	In the case where additional power is required for full operation the Higher Capability bit shall be set.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1 Sink Fixed Supply Power Data Object			
6.4.1.3.1#1	The Sink shall set Voltage to its required voltage and Operational Current to its required operating current.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1#2	All other Fixed Supply Power Supply objects shall set bits 29...20 to zero.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1#3	For a Sink requiring no power from the Source, the Voltage (B19..10) shall be set to 5V and the Operational Current shall be set to 0mA.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1#4	Bits B24_20 shall be set to zero	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1.1 Dual_Role Power			
6.4.1.3.1.1#1	The Dual_Role bit shall be set when the port is Dual_Role Power capable i.e. supports the PR_Swap message.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1.1#2	This is a static capability which shall remain fixed for a given device.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1.2 Higher Capability			
6.4.1.3.1.2#1	In the case that the Sink needs more than vSafe5V (e.g. 12V) to provide full functionality, then the Higher Capability bit shall be set.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1.3 Externally Powered			
6.4.1.3.1.3#1	The Externally Powered bit shall only be set when an AC or other bulk "wall" supply is providing 100% of the sink's power.	PROT_EXT_PWR	
6.4.1.3.1.4 USB Communications Capable			
6.4.1.3.1.4#1	The USB Communications Capable bit shall only be set for devices capable of communication over the USB data lines (e.g. D+/_ or SS Tx/Rx).	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1.5 Data Role Swap			
6.4.1.3.1.5#1	The Data Role Swap bit shall be set when the Port is a Type_C DRP device (see [USBType_C1.0]) and supports the DR_Swap message.	PROT_MSG_DATA_SNK_CAP_8	
6.4.1.3.1.5#2	This is a static capability which shall remain fixed for a given device.	PROT_MSG_DATA_SNK_CAP_8	
6.4.1.3.2 Variable Supply (non battery)Power Data Object			
6.4.1.3.2#1	The voltage fields shall be set to the output voltage range that the Sink requires to operate.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.2#2	The Operational Current field shall be set to the operational current that the Sink requires at the given voltage range.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.2#3	The absolute voltage, including any voltage variation, shall not fall below the Minimum Voltage and shall not exceed the Maximum Voltage.	BMC_POW_SRC_LOAD_P_PC_2 BMC_POW_SRC_LOAD_CP_ACC_3	

R2.0 Assertion #	Description	Test	Comments
6.4.1.3.3 Battery Supply Power Data Object			
6.4.1.3.3#1	The voltage fields shall be set to the output voltage range that the Sink requires to operate.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.3#2	The Operational Power field shall be set to the operational power that the Sink requires at the given voltage range.	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.3#3	The absolute voltage, including any voltage variation, shall not fall below the Minimum Voltage and shall not exceed the Maximum Voltage.	BMC_POW_SRC_LOAD_P_PC_2 BMC_POW_SRC_LOAD_CP_ACC_3	
6.4.2 Request Message			
6.4.2#1	A Request message shall be sent by a Sink to request power, typically during the request phase of a power negotiation.	PROT_MSG_DATA_REQ	
6.4.2#2	The Request Data Object shall be returned by the Sink making a request for power.	PROT_MSG_DATA_REQ	
6.4.2#3	It shall be sent in response to the most recent Source Capabilities message (see Section 8.3.2.3).	PROT_MSG_DATA_REQ_5	
6.4.2#4	A Request message shall return one and only one Sink Request Power Data Object that shall identify the Power Data Object being requested.	PROT_MSG_DATA_REQ_1	
6.4.2#5	(Table 6_14,15,16,17) Reserved – shall be set to zero	PROT_MSG_DATA_REQ_3 PROT_MSG_DATA_REQ_8	
6.4.2.1 Object Position			
6.4.2.1#1	The value in the Object Position field shall indicate which object in the Source Capabilities message the RDO refers.	PROT_MSG_DATA_REQ_5	
6.4.2.2 GiveBack Flag			
6.4.2.2#1	The GiveBack flag shall be set to indicate that the Sink will respond to a GotoMin message by reducing its load to the Minimum Operating Current.	PROT_MSG_DATA_REQ_13 PROT_MSG_DATA_REQ_14	
6.4.2.3 Capability Mismatch			
6.4.2.3#1	In this case the Sink shall make a valid request from the offered capabilities and shall set the Capability Mismatch bit (see Section 8.2.5.2).	PROT_MSG_DATA_REQ PROT_SEQ_CAP_MIS	
6.4.2.3#2	The Object position field shall contain a reference to an object in the last received Source Capabilities message.	PROT_MSG_DATA_REQ_5	
6.4.2.3#3	The Operating Current/Power field shall contain a value which is less than or equal to the maximum current/power offered in the Source Capabilities message.	PROT_MSG_DATA_REQ_9 PROT_MSG_DATA_REQ_12 PROT_MSG_DATA_REQ_15	
6.4.2.3#4	if the Capability Mismatch bit is set to zero § The Maximum Operating Current/Power field shall contain a value less than or equal to the maximum current/power offered in the Source Capabilities message's PDO as referenced by the Object position field.	PROT_MSG_DATA_REQ_16	
6.4.2.3#5	if the Giveback flag is set to one i.e. there is a Minimum Operating Current/Power field: o The Minimum Operating Current/Power field shall contain a value less than or equal to the Operating Current/Power field.	PROT_MSG_DATA_REQ_13 PROT_MSG_DATA_REQ_14	
6.4.2.4 USB Communications Capable			
6.4.2.4#1	The USB Communications Capable flag shall be set to one when the Sink has USB data lines and is capable of communicating using either [USB2.0] or [USB3.1] protocols.	PROT_MSG_DATA_REQ_6	
6.4.2.4#2	The USB Communications Capable flag shall be set to zero when the Sink does not have USB data lines or is otherwise incapable of communicating using either [USB2.0] or [USB3.1] protocols.		
6.4.2.6 Operating Current			
6.4.2.6#1	The Operating current field in the Request Data Object shall be set to the actual amount of current the Sink needs to operate at a given time.	PROT_MSG_DATA_REQ	
6.4.2.6#2	A new Request message, with an updated Operating Current		

R2.0 Assertion #	Description	Test	Comments
	value, shall be issued whenever the Sink's power needs change e.g. from Maximum Operating Current down to a lower current level.		
6.4.2.6#3	This field shall apply to the Fixed and Variable RDO.	PROT_MSG_DATA_REQ_12 PROT_MSG_DATA_REQ_15	
6.4.2.7 Maximum Operating Current			
6.4.2.7#1	The Maximum Operating Current field in the Request message shall be set to the highest current the Sink will ever require.	PROT_MSG_DATA_REQ_11 PROT_MSG_DATA_REQ_9	
6.4.2.7#2	The Operating Current value shall be less than or equal to the Maximum Operating Current value.	PROT_MSG_DATA_REQ_11 PROT_MSG_DATA_REQ_9	
6.4.2.7#3	When the Capabilities Mismatch bit is set to zero the requested Maximum Operating Current shall be less than or equal to the current in the offered Source Capabilities since the Source will need to reserve this power for future use.	PROT_MSG_DATA_REQ_16 PROT_MSG_DATA_REQ_10	
6.4.2.7#4	The Maximum Operating Current field shall continue to be set to the highest current needed in order to maintain the allocation of the Power Reserve.		
6.4.2.7#5	This field shall apply to the Fixed and Variable RDO.	PROT_MSG_DATA_REQ	
6.4.2.8 Minimum Operating Current			
6.4.2.8#1	The Minimum Operating Current field in the Request message shall be set to the lowest current the Sink requires to maintain operation.	PROT_MSG_DATA_REQ_13 PROT_MSG_DATA_REQ_14	
6.4.2.8#2	The Operating Current value shall be greater than or equal to the Minimum Operating Current value.	PROT_MSG_DATA_REQ_14	
6.4.2.8#3	This field shall apply to the Fixed and Variable RDO.	PROT_MSG_DATA_REQ	
6.4.2.9 Operating Power			
6.4.2.9#1	The Operating Power field in the Request Data Object shall be set to the actual amount of power the Sink wants at this time.	PROT_MSG_DATA_REQ_15 PROT_MSG_DATA_REQ_18	
6.4.2.9#2	This field shall apply to the Battery RDO.	PROT_MSG_DATA_REQ	
6.4.2.10 Maximum Operating Power			
6.4.2.10#1	The Maximum Operating Power field in the Request message shall be set to the highest power the Sink will ever require.	PROT_MSG_DATA_REQ_17	
6.4.2.10#2	The requested Maximum Operating Power shall be less than or equal to the power in the offered Source Capabilities since the Source will need to reserve this power for future use.	PROT_MSG_DATA_REQ_16	
6.4.2.10#3	The Maximum Operating Power field shall continue to be set to the highest power needed in order to maintain the allocation of the Power Reserve.	PROT_MSG_DATA_REQ_16	
6.4.2.10#4	This field shall apply to the Battery RDO.	PROT_MSG_DATA_REQ	
6.4.2.11 Minimum Operating Power			
6.4.2.11#1	The Minimum Operating Power field in the Request message shall be set to the lowest current the Sink requires to maintain operation.	PROT_MSG_DATA_REQ_19 PROT_MSG_DATA_REQ_20	
6.4.2.11#2	This field shall apply to the Battery RDO.	PROT_MSG_DATA_REQ	
6.4.3 BIST Message			
6.4.3#1	All ports shall be able to be a Unit Under Test (UUT) only when operating at vSafe5V.		
6.4.3#2	For each supported BIST Mode the following operations shall be implemented based on the reception of the appropriate BIST message Data Object (see Table 6_17): · Process reception of a BIST Receiver Mode Data Object		
6.4.3#3	For each supported BIST Mode the following operations shall be implemented based on the reception of the appropriate BIST message Data Object (see Table 6_17): · Process reception of a BIST Transmit Mode Data Object		

R2.0 Assertion #	Description	Test	Comments
6.4.3#4	For each supported BIST Mode the following operations shall be implemented based on the reception of the appropriate BIST message Data Object (see Table 6_17): Generate a Returned BIST Counters Data Object response within a BIST message in response to each received Test frame.		
6.4.3#5	For each supported BIST Mode the following operations shall be implemented based on the reception of the appropriate BIST message Data Object (see Table 6_17): · Process reception of a BIST Carrier Mode 0 Data Object that shall result in the generation of the appropriate carrier signal.		
6.4.3#6	For each supported BIST Mode the following operations shall be implemented based on the reception of the appropriate BIST message Data Object (see Table 6_17): · Process reception of a BIST Carrier Mode 1 Data Object that shall result in the generation of the appropriate carrier signal.		
6.4.3#7	For each supported BIST Mode the following operations shall be implemented based on the reception of the appropriate BIST message Data Object (see Table 6_17): · Process reception of a BIST Carrier Mode 2 Data Object that shall result in the generation of the appropriate carrier signal.	BMC_PHY_TX_BIT_2 BMC_PHY_TX_EYE_1 BMC_PHY_TX_EYE_2 CAB_PHY_TX_EYE_1 CAB_PHY_TX_EYE_2 CAB_PHY_TX_BIT_1 CAB_PHY_TX_BIT_2 CAB_PHY_RX_BUSIDL_1 [PROT_PROC_BIST_TSTR_1]	
6.4.3#8	For each supported BIST Mode the following operations shall be implemented based on the reception of the appropriate BIST message Data Object (see Table 6_17): · Process reception of a BIST Carrier Mode 3 Data Object that shall result in the generation of the appropriate carrier signal.		
6.4.3#9	For each supported BIST Mode the following operations shall be implemented based on the reception of the appropriate BIST message Data Object (see Table 6_17): · Process reception of a BIST Eye Pattern Data Object that shall result in the generation of the appropriate carrier signal.		
6.4.3#10	For each supported BIST Mode the following operations shall be implemented based on the reception of the appropriate BIST message Data Object (see Table 6_17): _ Process reception of a BIST Test Data Data Object, that shall result in the message being ignored.	BMC_PHY_RX_BUSIDL_1 BMC_PHY_RX_INT_REJ_1 BMC_PHY_TERM_1 CAB_PHY_RX_BUSIDL_1 CAB_PHY_RX_INT_REJ_1 CAB_PHY_TERM_1 [PROT_PROC_BIST_TSTR_1]	
6.4.3#11	When a Port receives a BIST message Data Object for a BIST Mode when Power Role swapped or not operating at vSafe5V, the BIST message shall be ignored.		
6.4.3#12	When a Port receives a BIST message Data Object for a BIST Mode it does not support the BIST message shall be ignored.		
6.4.3#13	When a Port or Cable Plug receives a BIST Message BIST Data Object for a Continuous BIST Mode that it supports, the Port or Cable Plug enters the requested BIST Mode and shall remain in that BIST Mode for tBISTContMode and then shall return to normal operation (see Section 6.5.8.4).		
6.4.3#14	a Tester shall always be able to complete the operations required when a BIST message with Data Object BIST Transmit Mode is sent by the Tester.	not tested	
6.4.3#15	A Tester Port shall have a means to place the UUT Port into	not tested	

R2.0 Assertion #	Description	Test	Comments
	receiver test mode and retrieve the error counters from the UUT.		
6.4.3#16	(Table 6_17 B15..0) When Request Type is Returned BIST Counters, this field shall contain the contents of BISTErrorCounter otherwise it shall be set to zero.		
6.4.3#17	(Table 6_17) Reserved and shall be set to zero.		
6.4.3#18	BIST messages shall be ignored unless VBUS is at vSafe5V and an Explicit contract is in place.		
6.4.3.1 BIST Receiver Mode			
6.4.3.1#1	This operation shall be used to initiate a UUT remote receiver test by sending a BIST message containing a BIST Receiver Mode BIST Data object.	PHYFSK_RX_SQU PHYFSK_RX_BER	
6.4.3.1#2	On receiving the request, the UUT shall zero its BISTErrorCounter and both the Tester and the UUT shall preload their PRBS generator with the designated pattern (see Section 5.9.1).	PHYFSK_RX_SQU PHYFSK_RX_BER	
6.4.3.1#3	The receiver (UUT) shall acknowledge the BIST message with a GoodCRC message.	PHYFSK_RX_SQU PHYFSK_RX_BER	
6.4.3.1#4	the UUT shall be able to receive a Test Frame from the Tester and to respond appropriately with a BIST message with a Data Object of Returned BIST Counters (see Section 6.5.2).		
6.4.3.1#5	The test shall be ended by sending Hard Reset signaling to reset the UUT.	PHYFSK_RX_SQU PHYFSK_RX_BER	
6.4.3.2 BIST Transmit Mode			
6.4.3.2#1	Loopback mode is not possible so BIST Transmit mode shall be used to request a UUT transmitter test by sending a 14 BIST message containing a BIST Transmit Mode BIST Data object.	PHYFSK_TX_BER	
6.4.3.2#2	Before initiating the request the Tester shall zero its BISTErrorCounter and preload the PRBS generator with the designated pattern (see Section 5.9.1).	PHYFSK_TX_BER	
6.4.3.2#3	On receiving the request the UUT shall preload the PRBS generator with the designated pattern (see Section 5.9.1).	PHYFSK_TX_BER	
6.4.3.2#4	The receiver in the UUT shall acknowledge the BIST message with a GoodCRC message.	PHYFSK_TX_BER	
6.4.3.2#5	The UUT shall start transmitting the 5 first Test Frame no later than tBISTMode max of the last bit of the EOP of the BIST message used to initiate the test is 6 received by the Physical Layer.		
6.4.3.2#6	Each subsequent Test Frame shall be started: · Either on reception of a BIST message, with a Returned BIST Counters Data Object or · On expiry of the BISTReceiveErrorTimer and · No later than tBISTResponse after the first bit of the Preamble of the previous Test Frame has been transmitted on VBUS by the Physical Layer.		
6.4.3.2#7	The Tester shall preload its PRBS checker with the designated pattern and start counting errors.	not tested	
6.4.3.2#8	After receiving a suitable number of test frames, the Tester shall freeze its error counter.	PROT_MSG_DATA_BIST	
6.4.3.2#9	The UUT shall be reset by sending Hard Reset signaling instead of a BIST message.	PHYFSK_TX_BER	
6.4.3.3 Returned BIST Counters			
6.4.3.3#1	The BIST message, with a Returned BIST Counters Data Object, shall contain the error counters obtained during the receiver test.	PHYFSK_RX_SQU PHYFSK_RX_BER	
6.4.3.3#2	During BIST, when sending Test Frames, the MessageID of the BIST message, with a Returned BIST Counters Data	not tested	

R2.0 Assertion #	Description	Test	Comments
	Object shall be ignored.		
6.4.3.4 BIST Carrier Mode 0			
6.4.3.4#1	Upon receipt of a BIST message, with a BIST Carrier Mode 0 Data Object, the UUT shall send out a continuous string of "0"s.	PHYFSK_TX_LEV	
6.4.3.4#2	The UUT shall exit the Continuous BIST Mode within tBISTContMode of this Continuous BIST Mode being enabled (see Section 6.5.8.4).	PHYFSK_TX_LEV	
6.4.3.5 BIST Carrier Mode 1			
6.4.3.5#1	Upon receipt of a BIST message, with a BIST Carrier Mode 1 Data Object, the UUT shall send out a continuous string of "1"s.	PHYFSK_TX_LEV	
6.4.3.5#2	The UUT shall exit the Continuous BIST Mode within tBISTContMode of this Continuous BIST Mode being enabled (see Section 6.5.8.4).	PHYFSK_TX_LEV	
6.4.3.6 BIST Carrier Mode 2			
6.4.3.6#1	Upon receipt of a BIST message, with a BIST Carrier Mode 2 Data Object, the UUT shall send out a continuous string of alternating "1"s and “0”s.	BMC_PHY_TX_EYE_2 CAB_PHY_TX_EYE_2 [PROT_PROC_BIST_TSTR_1]	
6.4.3.6#2	The UUT shall exit the Continuous BIST Mode within tBISTContMode of this Continuous BIST Mode being enabled (see Section 6.5.8.4).	BMC_PHY_TX_EYE_4 CAB_PHY_TX_EYE_4 CAB_PHY_TX_BIT_5 [PROT_PROC_BIST_TSTR_1]	
6.4.3.7 BIST Carrier Mode 3			
6.4.3.7#1	Upon receipt of a BIST message, with a BIST Carrier Mode 3 Data Object, the UUT shall send out a continuous string of alternating "1"s and “0”s.	PHYFSK_TX_FREQ	
6.4.3.7#2	The UUT shall exit the Continuous BIST Mode within tBISTContMode of this Continuous BIST Mode being enabled (see Section 6.5.8.4).	PHYFSK_TX_FREQ	
6.4.3.8 BIST Eye Pattern Test Mode			
6.4.3.8#1	Upon receipt of a BIST message, with a BIST Eye Pattern Data Object, the UUT shall send out a continuous string of bits in accordance with section 5.12.1.	PHYFSK_TX_EYE PHYFSK_TX_SPECT	
6.4.3.8#2	The UUT shall exit the Continuous BIST Mode within tBISTContMode of this Continuous BIST Mode being enabled (see Section 6.5.8.4).	PHYFSK_TX_EYE PHYFSK_TX_SPECT	
6.4.3.9 BIST Test Data			
6.4.3.9#1	Upon receipt of a BIST message, with a BIST Test Data Data Object, the UUT shall return a GoodCRC message and shall enter a test mode in which it sends no further messages except for GoodCRC messages in response to received messages.	BMC_PHY_RX_BUSIDL_2 BMC_PHY_TERM_1 BMC_PHY_MSG_1 CAB_PHY_RX_BUSIDL_2 CAB_PHY_TERM_1 PROT_PROC_BIST_TSTR_1	
6.4.3.9#2	The test shall be ended by sending Hard Reset signaling to reset the UUT.	PROT_PROC_BIST_TSTR_1	
6.4.4 Vendor Defined Message			
6.4.4#1	A Vendor Defined message shall consist of at least one data object, the VDM Header	PROT_MSG_DATA_VEND_1	
6.4.4#2	To ensure vendor uniqueness of Vendor Defined messages, all Vendor Defined messages shall contain a valid USB Standard or Vendor ID (SVID) allocated by USB_IF in the VDM Header.	PROT_MSG_DATA_VEND_2	
6.4.4#4	Vendor Defined messages shall not be used for direct power negotiation.		
6.4.4#5	The VDM header shall be the first 4_byte object in a Vendor Defined Message.	PROT_MSG_DATA_VDM_ID_INIT PROT_MSG_DATA_VDM_ID_ACK	
6.4.4#6	The fields in the VDM Header for an Unstructured VDM,		

R2.0 Assertion #	**Description**	**Test**	**Comments**
	when the VDM Type Bit is set to zero, shall be as defined in Table 6_18.		
6.4.4#7	The fields in the VDM Header for a Structured VDM, when the VDM Type Bit is set to one shall be as defined in Table 6_19.	PROT_MSG_DATA_VDM_ID_INIT_4 PROT_MSG_DATA_VDM_ID_ACK_4 PROT_MSG_DATA_VDM_SVID_INIT_4 PROT_MSG_DATA_VDM_SVID_ACK_4 PROT_MSG_DATA_VDM_MODE_INIT_4 PROT_MSG_DATA_VDM_MODE_ACK_4 PROT_MSG_DATA_VDM_ENTER_MODE_4 PROT_MSG_DATA_VDM_EXIT_MODE_4 PROT_MSG_DATA_VDM_ATT_4	
6.4.4#8	Both Unstructured and Structured VDMs shall only be sent and received after an Explicit PD Contract has been established.		
6.4.4#9	A VDM message sequence shall not interrupt any other PD Message Sequence.		
6.4.4#10	A VDM message sequence shall be interruptible by any other PD Message Sequence.		
6.4.4#11	The message format shall be as follows:	PROT_MSG_DATA_VEND	
6.4.4.1 Unstructured VDM			
6.4.4.1#1	The Port Partners and Cable Plugs shall exit any states entered using an Unstructured VDM when a Hard Reset appears on PD.		
6.4.4.1#2	The following rules apply to the use of Unstructured VDM messages: · Unstructured VDMs shall only be used when an Explicit Contract is in place.		
6.4.4.1#3	The following rules apply to the use of Unstructured VDM messages: · Prior to establishing an Explicit PD Contract Unstructured VDMs shall not be sent and shall be ignored if received.		
6.4.4.1#4	"The following rules apply to the use of Unstructured VDM Messages: A Port receiving an Unstructured VDM for a VID that it does not recognize shall Ignore the Message."		
6.4.4.1#5	The following rules apply to the use of Unstructured VDM messages: · Only the DFP shall be an Initiator of Unstructured VDMs.		
6.4.4.1#6	The following rules apply to the use of Unstructured VDM messages: · Only the UFP or a Cable Plug shall be a Responder to Unstructured VDM.		
6.4.4.1#7	The following rules apply to the use of Unstructured VDM messages: · Unstructured VDMs shall not be initiated or responded to under any other circumstances.		
6.4.4.1#8	The following rules apply to the use of Unstructured VDM messages: · A DFP or UFP which does not support Unstructured VDMs shall ignore any Unstructured VDMs received.		
6.4.4.1#9	The following rules apply to the use of Unstructured VDM messages: · A “command” sequence shall be interruptable e.g. due to the need for a message sequence using SOP Packets.		
6.4.4.1#10	The following rules apply to the use of Unstructured VDM messages: · Unstructured VDMs shall only be used during Modal Operation in the context of a Mode.		
6.4.4.1.1 USB Vendor ID			
6.4.4.1.1#1	The Vendor ID field shall contain the 16_bit Vendor ID value assigned to the vendor by the USB_IF (VID).		
6.4.4.1.1#2	No other value shall be present in this field.		

R2.0 Assertion #	Description	Test	Comments
6.4.4.1.2 VDM Type			
6.4.4.1.2#1	The VDM Type field shall be set to zero indicating that this is an Unstructured VDM.		
6.4.4.2 Structured VDM			
6.4.4.2#1	"Structured VDMs shall only be used when an Explicit Contract is in place with the following exception: o Prior to establishing an Explicit PD Contract a Source may issue Discover Identity messages, to a Cable Plug using SOP' Packets, as an Initiator (see Section 8.3.3.10.11)."	CAB_PROT_DISCOV_1 CAB_PROT_DISCOV_3 CAB_PROT_DISCOV_5 CAB_PROT_DISCOV_7 BMC_PROT_SEQ_CHKCAB_P_PC_1	
6.4.4.2#2	Only the DFP shall be an Initiator of Structured VDMs except for the Attention Command that shall only be initiated by the UFP.		
6.4.4.2#3	Only the UFP or a Cable Plug shall be a Responder to Structured VDMs		
6.4.4.2#4	Structured VDMs shall not be initiated or responded to under any other circumstances.		
6.4.4.2#5	A DFP or UFP which does not support Structured VDMs shall ignore any Structured VDMs received.		
6.4.4.2#6	A DFP, UFP or Cable Plug which supports Structured VDMs and receiving an Structured VDM for a SVID that it does not recognize shall reply with a NAK Command.	CAB_PHY_MSG_6	
6.4.4.2#7	A Command sequence shall be interruptable e.g. due to the need for a message sequesnce using SOP Packets.		
6.4.4.2#8	(Table 6_19) B12..11, B5 Reserved: shall be set to e0 and shall be ignored	PROT_MSG_DATA_VDM_ID_INIT_6 PROT_MSG_DATA_VDM_ID_INIT_9 PROT_MSG_DATA_VDM_ID_ACK_6 PROT_MSG_DATA_VDM_ID_ACK_9 PROT_MSG_DATA_VDM_SVID_INIT_6 PROT_MSG_DATA_VDM_SVID_INIT_9 PROT_MSG_DATA_VDM_SVID_ACK_6 PROT_MSG_DATA_VDM_SVID_ACK_9 PROT_MSG_DATA_VDM_MODE_INIT_6 PROT_MSG_DATA_VDM_MODE_INIT_9 PROT_MSG_DATA_VDM_MODE_ACK_6 PROT_MSG_DATA_VDM_MODE_ACK_9 PROT_MSG_DATA_VDM_ENTER_MODE_6 PROT_MSG_DATA_VDM_ENTER_MODE_9 PROT_MSG_DATA_VDM_EXIT_MODE_6 PROT_MSG_DATA_VDM_EXIT_MODE_9 PROT_MSG_DATA_VDM_ATT_6 PROT_MSG_DATA_VDM_ATT_9	
6.4.4.2#9	(0, 7_15) Reserved, shall not be used	PROT_MSG_DATA_VDM	
6.4.4.2#10	(Discover Identity, Discover SVIDs) Shall only use the PD_SID	PROT_MSG_DATA_VDM_ID_INIT_3 PROT_MSG_DATA_VDM_ID_ACK_3 PROT_MSG_DATA_VDM_SVID_INIT_3 PROT_MSG_DATA_VDM_SVID_ACK_3 PROT_MSG_DATA_VDM_MODE_INIT_3 PROT_MSG_DATA_VDM_MODE_ACK_3	
6.4.4.2#11	Table 6-20 shows the Commands, which SVID to use with each Command and the only SOP* values which shall be used.	PROT_MSG_DATA_VDM	
6.4.4.2#12	Table 6-20 Shall only use the PD SID (Discover Identity, Discover SVIDs)	PROT_MSG_DATA_VDM	
6.4.4.2#13	Table 6-20 Shall only use SOP/SOP' (Discover Identity, SVIDs, Modes)	PROT_MSG_DATA_VDM	
6.4.4.2.1 SVID			
6.4.4.2.1#1	The SVID field shall contain either a 16_bit USB Standard ID value (SID) or the 16_bit assigned to the vendor by the USB_IF (VID). No other value shall be present in this field.	PROT_MSG_DATA_VDM_ID_INIT_3 PROT_MSG_DATA_VDM_ID_ACK_3 PROT_MSG_DATA_VDM_SVID_INIT_3 PROT_MSG_DATA_VDM_SVID_ACK_3 PROT_MSG_DATA_VDM_MODE_INIT_3 PROT_MSG_DATA_VDM_MODE_ACK_3 PROT_MSG_DATA_VDM_ENTER_MODE_3 PROT_MSG_DATA_VDM_EXIT_MODE_3 PROT_MSG_DATA_VDM_ATT_3	
6.4.4.2.2 VDM Type			
6.4.4.2.2#1	The VDM type field shall be set to one indicating that this is	PROT_MSG_DATA_VDM	

R2.0 Assertion #	Description	Test	Comments
	a Structured VDM.		
6.4.4.2.3 Structured VDM Version			
6.4.4.2.3#1	This field shall be set to zero to indicate Version 1.0	PROT_MSG_DATA_VDM_ID_INIT_5 PROT_MSG_DATA_VDM_ID_ACK_5 PROT_MSG_DATA_VDM_SVID_INIT_5 PROT_MSG_DATA_VDM_SVID_ACK_5 PROT_MSG_DATA_VDM_MODE_INIT_5 PROT_MSG_DATA_VDM_MODE_ACK_5 PROT_MSG_DATA_VDM_ENTER_MODE_5 PROT_MSG_DATA_VDM_EXIT_MODE_5 PROT_MSG_DATA_VDM_ATT_5	
6.4.4.2.3#2	On receipt of a VDM message header with a higher Version number than supported, a port shall respond using the highest Version number it supports.		
6.4.4.2.4 Object Position			
6.4.4.2.4#1	The Object Position field shall be used by the Enter Mode and Exit Mode Commands.	PROT_MSG_DATA_VDM_ENTER_MODE_7 PROT_MSG_DATA_VDM_EXIT_MODE_7	
6.4.4.2.4#2	The Object Position shall start with one for the first Mode in the list.	CAB_PHY_MSG	
6.4.4.2.4#3	If the SVID is a VID, the content of the VDO for the Mode shall be defined by the vendor.		
6.4.4.2.4#4	If the SVID is a SID, the content shall be defined by the Standard.		
6.4.4.2.4#5	This field shall be set to zero when not required by the Command.	PROT_MSG_DATA_VDM_ID_INIT_7 PROT_MSG_DATA_VDM_ID_ACK_7 PROT_MSG_DATA_VDM_SVID_INIT_7 PROT_MSG_DATA_VDM_SVID_ACK_7 PROT_MSG_DATA_VDM_MODE_INIT_7 PROT_MSG_DATA_VDM_MODE_ACK_7	
6.4.4.2.5 Command Type			
6.4.4.2.5#1	This Command Type field shall be used to indicate the type of Command request/response being sent.	PROT_MSG_DATA_VDM_ID_INIT_8 PROT_MSG_DATA_VDM_ID_ACK_8 PROT_MSG_DATA_VDM_SVID_INIT_8 PROT_MSG_DATA_VDM_SVID_ACK_8 PROT_MSG_DATA_VDM_MODE_INIT_8 PROT_MSG_DATA_VDM_MODE_ACK_8 PROT_MSG_DATA_VDM_ENTER_MODE_8 PROT_MSG_DATA_VDM_EXIT_MODE_8 PROT_MSG_DATA_VDM_ATT_8	
6.4.4.2.5#2	An Initiator shall set the field to "Initiator" to indicate that this is a Command request from an Initiator.	PROT_MSG_DATA_VDM_ID_INIT_8 PROT_MSG_DATA_VDM_SVID_INIT_8 PROT_MSG_DATA_VDM_MODE_INIT_8 PROT_MSG_DATA_VDM_ENTER_MODE_8 PROT_MSG_DATA_VDM_EXIT_MODE_8 PROT_MSG_DATA_VDM_ATT_8	
6.4.4.2.5#3	“Responder ACK” is the normal return and shall be sent to indicate that the Command request was received and handled normally.	PROT_MSG_DATA_VDM	
6.4.4.2.5#4	“Responder NAK” shall be returned when the Command request has an invalid parameter (e.g. invalid SVID or Mode) or cannot not be acted upon at this time (e.g. a Mode which has a dependency on another Mode).	PROT_MSG_DATA_VDM	
6.4.4.2.5#5	“Responder BUSY” shall be sent in the response to a VDM when the Responder is unable to respond to the Command request immediately, but the Command request may be retried.	PROT_MSG_DATA_VDM	
6.4.4.2.5#6	The DFP shall wait at least tVDMBusy after a "Responder BUSY" response is received before retrying the Command request.	BMC_PROT_DISCOV CAB_PROT_DISCOV	
6.4.4.2.6 Command			
6.4.4.2.6#1	The commands defined in this specificaiton shall be used to manage Modes on the US Type_C connector.		
6.4.4.2.6#2	If the Structured VDM Command request is not recognized it shall be NAKed.		
6.4.4.3 Use of Commands			
6.4.4.3#1	All ports that support Modes shall support the Discover	CAB_PROT_DISCOV_1	

R2.0 Assertion #	Description	Test	Comments
	Identitiy, Discover SVIDs, the Discover Modes, the Enter Mode and Exit Mode Commands.	CAB_PROT_DISCOV_3 CAB_PROT_DISCOV_5 CAB_PROT_DISCOV_7	
6.4.4.3#2	Responses not listed for a given Command shall not be sent by a Responder.		
6.4.4.3.1 Discover Identity			
6.4.4.3.1#1	The Discover Identity Command shall be used to determine whether a given Cable Plug is PD Capable (see Sections 8.3.3.9.2 and 8.3.3.10.11).		
6.4.4.3.1#2	In this case a Discover Identity Command request sent to SOP' shall not cause a Soft Reset if a GoodCRC Message response is not returned since this can indicate a non-PD Capable cable.	BMC_PROT_SEQ_CHKCAB_P_PC_1	
6.4.4.3.1#3	A PD-Capable Cable Plug shall return a Discover Identity Command ACK in response to a Discover Identity Command request sent to SOP'.	CAB_PROT_DISCOV_1	
6.4.4.3.1#4	A PD-Capable UFP that supports Modal Operation shall return a Discover Identity Command ACK in response to a Discover Identity Command request sent to SOP.	BMC_PROT_DISCOV_1	
6.4.4.3.1#5	The SVID in the Discover Identity Command shall be set to the PD SID (see table 6_21) by both the Initiator and the Responder for this Command.	PROT_MSG_DATA_VDM_ID_INIT_3 PROT_MSG_DATA_VDM_ID_ACK_3	
6.4.4.3.1#6	No VDOs other than those defined in this specification shall be sent as part of the Discover Identity Command response.	PROT_MSG_DATA_VDM_ID_ACK_48	
6.4.4.3.1#7	Where there is no Product Type VDO defined for a specific Product Type, no VDOs shall be sent as part of the Discover Identity Command response.		
6.4.4.3.1#8	Any additional VDOs received by the responder shall be ignored.		
6.4.4.3.1.1 ID Header			
6.4.4.3.1.1#1	The fields in the ID Header shall be as defined in Table 6_23	PROT_MSG_DATA_VDM_ID_ACK	
6.4.4.3.1.1#2	Data Capable as USB Host: · Shall be set to one if the product is capable of enumerating USB Devices. · Shall be set to zero otherwise	PROT_MSG_DATA_VDM_ID_ACK_11	
6.4.4.3.1.1#3	Data Capable as a USB Device: · Shall be set to one if the product is capable of enumerating as a USB Device. · Shall be set to zero otherwise	PROT_MSG_DATA_VDM_ID_ACK_12	
6.4.4.3.1.1#4	B25…16: Shall be set to zero	PROT_MSG_DATA_VDM_ID_ACK_15	
6.4.4.3.1.1#5	Modal Operation Supported: · Shall be set to one if the product supports Modal Operation. · Shall be set to zero otherwise	PROT_MSG_DATA_VDM_ID_ACK_14	
6.4.4.3.1.4 Product Type			
6.4.4.3.1.4#1	Table 6_24 defines the Product Type VDOs which shall be returned.	PROT_MSG_DATA_VDM_ID_ACK_13	
6.4.4.3.1.4#2	Shall be used where no other Product Type value is appropriate.	PROT_MSG_DATA_VDM_ID_ACK	
6.4.4.3.1.4#3	Shall be used when the Product is a USB Hub.	PROT_MSG_DATA_VDM_ID_ACK	
6.4.4.3.1.4#4	Shall be used when the Product is a USB Device other than a USB Hub.	PROT_MSG_DATA_VDM_ID_ACK	
6.4.4.3.1.4#5	Shall be used when the Product is a cable that incorporates signal conditioning circuits.	PROT_MSG_DATA_VDM_ID_ACK	
6.4.4.3.1.4#6	Shall be used when the Product is a cable that does not incorporate signal conditioning circuits.	PROT_MSG_DATA_VDM_ID_ACK	
6.4.4.3.1.4#7	Shall be used when the Product is a USB Device that supports one or more Alternate Modes.	PROT_MSG_DATA_VDM_ID_ACK	
6.4.4.3.1.6 Vendor ID			

R2.0 Assertion #	Description	Test	Comments
6.4.4.3.1.6#1	Manufacturers shall set the Vendor ID field to the value of the Vendor ID assigned to them by USB_IF.	PROT_MSG_DATA_VDM_ID_ACK_16	
6.4.4.3.1.6#2	For USB Devices or Hubs which support USB communications the Vendor ID field shall be identical to the Vendor ID field defined in the product's USB Device Descriptor (see [USB2.0] and [USB3.1]).		
6.4.4.3.1.7 Cert Stat VDO			
6.4.4.3.1.7#1	The Cert Stat VDO shall contain the Test ID (TID) assigned by USB_IF to the product during certification.	PROT_MSG_DATA_VDM_ID_ACK_18	
6.4.4.3.1.7#2	Table 6_25 B31..20: Reserved, shall be set to zero.	PROT_MSG_DATA_VDM_ID_ACK_17	
6.4.4.3.1.8 Product VDO			
6.4.4.3.1.8#1	The fields in the Product VDO shall be as defined in Table 6_26.	PROT_MSG_DATA_VDM_ID_ACK_19 PROT_MSG_DATA_VDM_ID_ACK_20	
6.4.4.3.1.8#2	For USB Devices or Hubs which support USB Communications the Product ID and bcdDevice fields shall be identical to the Product ID and bcdDevice fields defined in the product's USB Device Descriptor (see [USB2.0] and [USB3.1]).		
6.4.4.3.1.9 Cable VDO			
6.4.4.3.1.9#1	The Cable VDO defined in this section shall be sent when the Product Type is given as Passive or Active Cable.	PROT_MSG_DATA_VDM_ID_ACK_49	
6.4.4.3.1.9#2	Table 6_27 defines the Cable VDO which shall be sent.	PROT_MSG_DATA_VDM_ID_ACK_22 PROT_MSG_DATA_VDM_ID_ACK_23 PROT_MSG_DATA_VDM_ID_ACK_25 PROT_MSG_DATA_VDM_ID_ACK_27 PROT_MSG_DATA_VDM_ID_ACK_28 PROT_MSG_DATA_VDM_ID_ACK_29 PROT_MSG_DATA_VDM_ID_ACK_30 PROT_MSG_DATA_VDM_ID_ACK_31 PROT_MSG_DATA_VDM_ID_ACK_32 PROT_MSG_DATA_VDM_ID_ACK_33 PROT_MSG_DATA_VDM_ID_ACK_34 PROT_MSG_DATA_VDM_ID_ACK_35 PROT_MSG_DATA_VDM_ID_ACK_36	
6.4.4.3.1.9#3	Table 6_27 B23..20 Reserved: Shall be set to zero.	PROT_MSG_DATA_VDM_ID_ACK_24	
6.4.4.3.1.10 Alternate Mode Adapter VDO			
6.4.4.3.1.10#1	The Alternate Mode Adapter (AMA) VDO in this section shall be sent when the Product Type is given as Alternate Mode Adapter.	PROT_MSG_DATA_VDM_ID_ACK_50	
6.4.4.3.1.10#2	Table 6_28 defines the AMA VDO which shall be sent.	PROT_MSG_DATA_VDM_ID_ACK_37 PROT_MSG_DATA_VDM_ID_ACK_38 PROT_MSG_DATA_VDM_ID_ACK_40 PROT_MSG_DATA_VDM_ID_ACK_41 PROT_MSG_DATA_VDM_ID_ACK_42 PROT_MSG_DATA_VDM_ID_ACK_43 PROT_MSG_DATA_VDM_ID_ACK_44 PROT_MSG_DATA_VDM_ID_ACK_45 PROT_MSG_DATA_VDM_ID_ACK_46 PROT_MSG_DATA_VDM_ID_ACK_47	
6.4.4.3.1.10#3	B23..12 Shall be set to zero.	PROT_MSG_DATA_VDM_ID_ACK_39	
6.4.4.3.2 Discover SVIDs			
6.4.4.3.2#1	The SVID in the Discover SVIDs Command shall be set to the PD SID (see Table 6_21) by both the Initiator and the Responder for this Command.	PROT_MSG_DATA_VDM_SVID_INIT_3	
6.4.4.3.2#2	A Responder shall only return SVIDs for which a Discover Modes Command request for that SVID will return at least one Mode		
6.4.4.3.2#3	A Responder that does not support any SVIDs shall return a NAK.	BMC_PROT_DISCOV_3	
6.4.4.3.2#4	If the Responder supports 12 or more SVIDs then the Discover SVIDs Command shall be executed multiple times until a Discover SVIDs VDO is returned ending either with a SVID value of 0x0000 in the last part of the last VDO or with	PROT_MSG_DATA_VDM_SVID_ACK_11 PROT_MSG_DATA_VDM_SVID_ACK_12 PROT_MSG_DATA_VDM_SVID_ACK_13	

R2.0 Assertion #	Description	Test	Comments
	a VDO containing two SVIDs with values of 0x0000.		
6.4.4.3.2#5	Each Discover SVID ACK message, other than the one containing the terminating 0x0000 SVID, shall convey 12 SVIDs.	BMC_PROT_DISCOV CAB_PROT_DISCOV	
6.4.4.3.2#6	The Responder shall restart the list of SVIDs each time a Discover Identity Command request is received from the Initiator.	BMC_PROT_DISCOV CAB_PROT_DISCOV	
6.4.4.3.3 Discover Modes			
6.4.4.3.3#1	The Responder to the Discover Modes command shall return a Message Header with the number of Data Objects field set to a value of 1 to 7 (the actual number is the number of Mode objects plus one).	PROT_MSG_DATA_VDM_MODE_ACK_1	
6.4.4.3.3#2	A Responder that does not support and Modes shall return a NAK.	CAB_PROT_DISCOV_5	
6.4.4.3.4 Enter Mode Command			
6.4.4.3.4#1	The value in the Object Position field in the VDM Header shall indicate to which Mode in the Discover Modes Command the VDO refers (see Figure 6-9).	PROT_MSG_DATA_VDM_ENTER_MODE_7	
6.4.4.3.4#2	The Number of Data Objects field in the Message Header in the Command request shall be 1 set to either 1 or 2.	PROT_MSG_DATA_VDM_ENTER_ MODE_1	
6.4.4.3.4#3	The Enter Mode Command request shall not contain more than 1 VDO.	PROT_MSG_DATA_VDM_ ENTER_MODE_11	
6.4.4.3.4#4	The Number of Data Objects field in the Command response shall be set to 1 since an Enter Mode Command response (ANK, NAK, BUSY) shall not contain any VDOs.	PROT_MSG_DATA_VDM_ENTER_MODE_1 PROT_MSG_DATA_VDM_ENTER_MODE_11	
6.4.4.3.4#5	Before entering a Mode, by sending the Enter Mode Command request, that requires the reconfiguring of any pins on entry to that Mode, the Initiator shall ensure that those pins being reconfigured are placed into the USB Safe State.		
6.4.4.3.4#6	Before entering a Mode that requires the reconfiguring of any pins, the Responder shall ensure that those pins being reconfigured are placed into either USB operation or the USB Safe State.		
6.4.4.3.4#7	Where there are multiple Active Modes are active at the same time, Modal operation shall start on entry to the first Mode.		
6.4.4.3.4#8	On receiving the Enter Mode Command request the Responder shall respond with either an ACK or a NAK response.	CAB_PROT_DISCOV_7	
6.4.4.3.4#9	The value in the Object Position field of the Enter Mode Command response shall contain the same value as the received Enter Mode Command request.	PROT_MSG_DATA_VDM_ ENTER_MODE_7	
6.4.4.3.4#10	If the Responder responds to the Enter Mode Command request with an ACK, the Responder shall enter the Mode before sending the ACK.		
6.4.4.3.4#11	The Initiator shall enter the Mode on reception of the ACK.		
6.4.4.3.4#12	If the Responder responds to the Enter Mode Command request with a NAK, the Mode is not entered. If not presently in Modal operation the Initiator shall return to USB operation.		
6.4.4.3.4#13	If the Responder responds to the Enter Mode Command with a NAK, the Mode is not entered…. If not presently in Modal operation the Responder shall remain in either USB operation or the USB Safe State.		
6.4.4.3.4#14	If the Initiator fails to receive a response within tVDMWaitModeEntry it shall not enter the Mode but return to USB operation.		
6.4.4.3.4#15	Once the Mode is entered, the device shall remain in that	CAB_PROT_DISCOV_8	

R2.0 Assertion #	Description	Test	Comments
	Active Mode until the Exit Mode command is successful (see Section 6.4.4.3.5)		
6.4.4.3.4#16	The following events shall also cause the Port Partners and Cable Plug(s) to exit of all presently active Modes: _ A PD Hard Reset _ The Port Partners or Cable Plug(s) are disconnected _ A Data Role Swap _ A Cable Reset		
6.4.4.3.4#17	The Initiator shall return to USB Operation within tVDMExitMode fo a disconnect or of Hard Reset signaling being detected.		
6.4.4.3.4#19	The Responder shall return to either USB Operation or USB Safe State within tVDMExitMode of a disconnect or of Hard Reset signaling detected.		
6.4.4.3.4#20	A DR_Swap message shall not be sent during Modal Operation.		
6.4.4.3.5 Exit Mode Command			
6.4.4.3.5#1	The value in the Object Position field shall indicate to which Mode in the Discover Modes Command the VDO refers (see Figure 6-10) and the shall have been used previously in an Enter Mode Command request for a Active Mode.	PROT_MSG_DATA_VDM_EXIT_MODE_7	
6.4.4.3.5#2	A value of 111b in the Object Position field shall indicate that all Active Modes shall be exited.		
6.4.4.3.5#3	The Number of Data Objects field in both the Command Request and th Command Response shall be set to 1 since an Exit Mode Command shall not contain any VDOs.	PROT_MSG_DATA_VDM_EXIT_MODE_1 PROT_MSG_DATA_VDM_EXIT_MODE_11	
6.4.4.3.5#4	The Responder shall exit its Active Mode before sending the response message.		
6.4.4.3.5#5	The Initiator shall exit its Active Mode before senting GoodCRC message in response to the ACK.		
6.4.4.3.5#6	The Responder shall not return a BUSY acknowledgement and shall only return a NAK acknowledgement to a request not containing an Active Mode (i.e. invalid object position).		
6.4.4.3.5#7	An Initiator which fails to receive an ACK within tVDMWaitModeExit or receives a NAK or BUSY response shall exit its Active Mode.		
6.4.4.3.6 Attention			
6.4.4.3.6#1	The value in the Object Position field shall indicate to which Mode in the Discover Modes Command the VDO refers (see Figure 6-9) and the shall have been used previously in an Enter Mode Command request for an Active Mode.	PROT_MSG_DATA_VDM_ATT_7	
6.4.4.3.6#2	A value of 000b or 111b in the Object Position field shall not be used by the Attention Command.		
6.4.4.3.6#3	The Number of Data Objects field in the Message Header shall be set to 1 or 2.	PROT_MSG_DATA_VDM_ATT_1 PROT_MSG_DATA_VDM_ATT_11	
6.4.4.3.6#4	The Attention Command shall not contain more than 1 VDO.		
6.4.4.3.6#5	No more than nAttentionCount Commands shall be sent during any given tAttentionAverage (max) period.		
6.4.4.3.6#6	The spacing between successive Attention Commands shall be at least tAttentionSpacing except that a single burst of no more than 2 Attention Commands with a spacing of at least tAttentionBurstSpacing may be sent during any given tAttentionAverage (max) period.		
6.4.4.4 Command Process			
6.4.4.4#1	The Responder shall complete: · Enter Mode requests within tVDMEnterMode · Exit Mode requests within tVDMExitMode · Other requests within tVDMReceiverResponse,	CAB_PROT_DISCOV BMC_PROT_DISCOV	

R2.0 Assertion #	Description	Test	Comments
6.4.4.4#2	An Initiator not receiving a response within the following times shall timeout and return to either the PE_SRC_Ready or PE_SNK_Ready state (as appropriate): · Enter Mode requests within tVDMWaitModeEntry · Exit Mode requests within tVDMWaitModeExit · Other requests within tVDMSenderResponse,		
6.4.4.4#3	"The Responder shall respond with: ACK if it recognizes the SVID and can process it at this time NAK o if it recognizes the SVID but cannot process the command request o or if it does not recognize the SVID o or if it does support the command request o or if a VDO has been supplied which is invalid. BUSY if it recognizes the SVID and the command but cannot process the Command request at this time"	CAB_PROT_DISCOV_3	
6.4.4.4.2 Enter Vendor Mode/ Exit Vendor Mode Processes			
6.4.4.4.2#1	In a managed termination, using the Exit Mode command, the Active Mode shall be exited in a controlled manner as described in Section 6.4.4.3.5.		
6.4.4.4.2#2	In an unmanaged termination, triggered by a Power Delivery Hard Reset (i.e. Hard Reset signaling sent by either Port Partner) or a Data Role swap or by cable detach (device unplugged), the Active Mode shall still be exited but there may not be a transition through the USB Safe State.		
6.4.4.5 VDM message Timing and Normal PD messages			
6.4.4.5#1	The Vendor or Standards defined state operation shall comprehend this and continue to operate as expected when processing any other USB PD Messages.		
6.4.4.5#2	The timing and interspersing of VDMs wetween regular PD messages shall be done without perturbing the PD message sequences.		
6.4.4.5#3	This requirement shall apply to both Unstructured VDMs and Structured VDMs.		
6.4.4.5#4	Structured and Unstructured VDM sequences shall not be atomic although they may require a strict albeit interrupted ordering.		
6.4.4.5#5	The use of Structured VDMs by a Initiator shall not interfere with the normal PD message timing requirements nor shall either the Initiator or Responder interrupt a PD message sequence.		
6.4.4.5#6	The Use of Unstructured VDMs shall not interfere with normal PD message timing.		
6.4.4.5#7	VDM sequences shall be interruptible after the return of a GoodCRC message has been completed.		
6.5.1 CRCReceiveTimer			
6.5.1#1	The CRCReceiveTimer shall be used by the sender's Protocol Layer to ensure that a message has not been lost.		
6.5.1#2	The sender’s Protocol Layer response when a CRCReceiveTimer expires shall be to retry nRetryCount times.		
6.5.1#3	Sending of the Preamble corresponding to the retried Message shall start within tRetry of the CRCReceiveTimer expiring.		
6.5.1#4	The CRCReceiveTimer shall be started when the last bit of the message EOP has been transmitted by the Physical Layer.		
6.5.1#5	The CRCReceiveTimer shall be stopped when the last bit of the EOP corresponding to the GoodCRC message has been received by the Physical Layer.		

R2.0 Assertion #	Description	Test	Comments
6.5.1#6	The Protocol Layer receiving a message shall respond with a GoodCRC message within tTransmit in order to ensure that the sender's CRCReceiveTimer does not expire.	PROT_MSG_CTRL PROT_PROC_GOODCRC_UUT_1	
6.5.1#7	The tTransmit shall be measured from when the last bit of the message EOP has been received by the Physical Layer until the first bit of the preamble of the GoodCRC message has been transmitted by the Physical Layer.	PROT_MSG_CTRL PROT_PROC_GOODCRC_UUT_1	
6.5.2 SenderResponseTimer			
6.5.2#1	The SenderResponseTimer shall be used by the sender's Policy Engine to ensure that a message requesting a response (i.e. Get_Source_Cap message) is responded to within a bounded time of tSenderResponse.		
6.5.2#2	The Policy Engine's response when the SenderResponseTimer expires shall be dependent on the message sent (see Section 8.3).		
6.5.2#3	The SenderResponseTimer shall be started from the time the last bit of the GoodCRC message EOP (i.e. the GoodCRC message corresponding to the message requesting a response) has been received by the Physical Layer.		
6.5.2#4	The SenderResponseTimer shall be stopped when the last bit of the expected response message EOP has been received by the Physical Layer.		
6.5.2#5	The receiver of a message requiring a response shall respond within tReceiverResponse in order to ensure that the sender's SenderResponseTimer does not expire.	PROT_PROC_SWAP_TSTR_SNK_1	
6.5.2#6	The tReceiverResponse time shall be measured from the time the last bit of the message EOP has been received by the Physical Layer until the first bit of the response message preamble has been transmitted by the Physical Layer.	PROT_MSG_CTRL	
6.5.3.1 SourceActivityTimer			
6.5.3.1#1	In order to maintain bus activity the Source shall start its SourceActivityTimer as described in Section 8.3.3.2.	PROT_PROC_SWAP_TSTR_SNK_4 PROT_PROC_SWAP_TSTR_SRC_5 PROT_PROC_SWAP_UUT_SNK_2 PROT_PROC_SWAP_UUT_SNK_7 PROT_PROC_SWAP_UUT_SRC_2 PROT_PROC_SWAP_UUT_SRC_5 PROT_PROC_SWAP_UUT_SRC_7 PROT_PROC_REQ_TSTR	
6.5.3.1#2	Whenever the timer expires, after tSourceActivity, the Source shall send a Ping message.	PROT_MSG_CTRL_PING_2 PROT_PROC_SWAP_TSTR_SNK_4 PROT_PROC_SWAP_TSTR_SRC_5 PROT_PROC_SWAP_UUT_SNK_2 PROT_PROC_SWAP_UUT_SRC_5 PROT_PROC_SWAP_UUT_SRC_2 PROT_PROC_SWAP_UUT_SRC_7 PROT_PROC_REQ_TSTR PROT_PROC_PING_2	
6.5.3.1#3	It shall then initialize and restart the SourceActivityTimer ready for the next Ping message.	PROT_MSG_CTRL_PING_2 PROT_PROC_SWAP_TSTR_SNK_4 PROT_PROC_SWAP_TSTR_SRC_5 PROT_PROC_SWAP_UUT_SNK_2 PROT_PROC_SWAP_UUT_SRC_7 PROT_PROC_SWAP_UUT_SRC_5 PROT_PROC_SWAP_UUT_SRC_2 PROT_PROC_REQ_TSTR	
6.5.3.1#4	The SourceActivityTimer shall be reset and restarted on entry to the PE SRC Ready state, as defined in Section 8.3.3.2, when: • The last bit of the GoodCRC message EOP has been received on VBUS by the Physical Layer. (i.e. a message has been successfully sent prior to entering the Ready state).		
6.5.3.1#5	The SourceActivityTimer shall be reset and restarted on		

R2.0 Assertion #	Description	Test	Comments
	entry to the Ready state, as defined in Section 8.3.3.2, when: • The last bit of any message EOP has been received on VBUS by the Physical Layer.		
6.5.3.1#6	The SourceActivityTimer shall be reset and restarted on entry to the Ready state, as defined in Section 8.3.3.2, when: • The SenderResponseTimer times out.		
6.5.3.1#7	This shall cause the Source to send a Soft Reset message, transmission of which shall be completed within tSoftReset of the CRCReceiveTimer expiring		
6.5.3.2 SinkActivityTimer			
6.5.3.2#1	The Sink shall support the SinkActivityTimer.		
6.5.3.2#2	Sinks shall observe an absence of a Ping, or other messages within tSinkActivity, as an indication of communications failure and as such shall issue Hard Reset signaling in order to return to default operation.		
6.5.3.2#3	Sinks shall also use the absence of VBUS to return to default operation unless this is part of an ongoing Power Role Swap sequence.		
6.5.3.2#4	The SinkActivityTimer shall be re_initialized and re_started when the last bit of a Ping or any other message EOP has been received by the Physical Layer.		
6.5.3.2#5	"to avoid triggering unnecessary Hard Resets the SinkActivityTimer shall not be run when the Sink is: _Using a [USBType_C1.0] connector or _Using Type_A operating in its default Source role or Type_B connector operating in its default role, at vSafe5V, in the PE_SNK_Ready state. This is because Ping messages are optional in these cases, (see Section 6.3.5)."	not tested	
6.5.4.1 SourceCapabilityTimer			
6.5.4.1#1	Prior to a successful negotiation a Source shall use the SourceCapabilityTimer to periodically send out a Source Capabilities message every tTypeCSendSourceCap for the Type_C connector and tSendSourceCap for all other connectors while: · An A_plug is attached. · The Source is not in an active connection with a PD Sink Port.		
6.5.4.1#2	Whenever there is a SourceCapabilityTimer timeout the Source shall send a Source Capabilities message.		
6.5.4.1#3	It shall then re_initialize and restart the SourceCapabilityTimer.		
6.5.4.1#4	The SourceCapabilityTimer shall be stopped when the last bit of the EOP corresponding to the GoodCRC message has been received by the Physical Layer since a PD connection has been established.		
6.5.4.2 SinkWaitCapTimer			
6.5.4.3#1	The Sink shall support the SinkWaitCapTimer.		
6.5.4.3#2	When a Sink observes an absence of Source Capabilities messages, after VBUS is present, for a duration of tTypeCSinkWaitCap for the Type_C connector and tSinkWaitCap for all other connectors the Sink shall issue Hard Reset signaling in order to restart the sending of Source Capabilities messages by the Source (see Section 6.6.4).		
6.5.4.3 tFirstSourceCap			
6.5.4.3#1	After Port Partners are attached or after a Hard Reset or after a Power Role Swap a Source shall send its first Source_Capabilities Message within tFirstSourceCap of VBUS reaching vSafe5V.	[PROT_PROC_HR_TSTR_5]	
6.5.5 Sink Request Timer			
6.5.5#1	The SinkRequestTimer shall be started when a Wait message		

R2.0 Assertion #	Description	Test	Comments
	has been received…		
6.5.5#2	The SinkRequestTimer...shall be stopped if any other message is received or during a Hard Reset.		
6.5.5#3	The Sink shall wait at least tSinkRequest, after receiving a Wait message, before issuing a new Request message.		
6.5.5#4	It shall then re_initialize and restart the SinkRequestTimer.		
6.5.6.1 PSTransition Timer			
6.5.6.1#1	The PSTransitionTimer shall be started when the last bit of an Accept or GotoMin message EOP has been receivedby the Physical Layer.	PROT_HS_PSTRANS_C_CP PROT_HS_PSTRANS_PC	
6.5.6.1#2	The PSTransitionTimer shall be stopped when the last bit of the PS_RDY message EOP has been received by the Physical Layer.	PROT_HS_PSTRANS_C_CP PROT_HS_PSTRANS_PC	
6.5.6.2 PSSourceOffTimer			
6.5.6.2#1a	If a PR Swap message request, has been sent by the Dual_Role Device currently acting as a Source the Sink can respond with an Accept message. When the last bit of the EOP of the GoodCRC message corresponding to this Accept message is received by the Sink, then the PSSourceOffTimer shall be started.	PROT_PROC_PSSOURCEOFFTIMER_1 PROT_PROC_PSSOURCEOFFTIMER_SWPD_1	
6.5.6.2#1b	If a PR Swap message request, has been sent by the Dual_Role Device currently acting as a Sink the Source can respond with an Accept message. When the last bit of the EOP of this Accept message is received by the Sink then the PSSourceOffTimer shall be started.	PROT_PROC_PSSOURCEOFFTIMER_1 PROT_PROC_PSSOURCEOFFTIMER_SWPD_1	
6.5.6.2#2	The PSSourceOffTimer shall be stopped when: · The last bit of the EOP of the PS_RDY message is received.	PROT_PROC_PSSOURCEOFFTIMER_1 PROT_PROC_PSSOURCEOFFTIMER_SWPD_1	
6.5.6.2#3	The timer shall time out if a PS_RDY message has not been received from the remote Dual_Role Device within tPSSourceOff indicating this has occurred.	PROT_PROC_PSSOURCEOFFTIMER_1 PROT_PROC_PSSOURCEOFFTIMER_SWPD_1 PROT_PROC_SWAP_TSTR_SNK_5	
6.5.6.3 PSSourceOnTimer			
6.5.6.3#1	The PSSourceOnTimer shall be started when: · The last bit of the EOP of the Good CRC message corresponding to the transmitted PS_RDY message is received by the Physical Layer.	PROT_PROC_PSSOURCEONTIMER_1 PROT_PROC_PSSOURCEONTIMER_SWPD_1 PROT_PROC_SWAP_TSTR_SRC_6	
6.5.6.3#2	The PSSourceOnTimer shall be stopped when: · The last bit of the EOP of the PS_RDY message is received by the Physical Layer.	PROT_PROC_PSSOURCEONTIMER_1 PROT_PROC_PSSOURCEONTIMER_SWPD_1	
6.5.7 NoResponseTimer			
6.5.7#1	When the NoResponseTimer times out, the Policy Engine shall issue up to nHardResetCount additional Hard Resets before determining that the Port Partner is non_responsive to USB Power Delivery messaging.		
6.5.7#2	If the Sink fails to receive a Capabilities message received within tNoResponse of: The last bit of a Hard Reset signaling being sent by the PHY layer if the Hard Reset signaling was initiated by the Sink. · The last bit of a Hard Reset signaling being received by the PHY layer if the Hard Reset signaling was initiated by the Source. Then the Sink shall issue additional Hard Resets up to nHardResetCount times (see Section 6.7.2).		
6.5.7#3	If the Source fails to receive a GoodCRC message in response to a Capabilities message within tNoResponse of: The last bit of a Hard Reset signaling being sent by the PHY layer if the Hard Reset signaling was initiated by the Sink. · The last bit of a Hard Reset signaling being received by the PHY layer if the Hard Reset signaling was initiated by the Source. Then the Source shall issue additional Hard Resets up to		

R2.0 Assertion #	Description	Test	Comments
	nHardResetCount times (see Section 6.7.2).		
6.5.8.1 tBISTMode			
6.5.8.1#1	A UUT shall enter the appropriate BIST mode and shall send the last bit of the EOP of the GoodCRC message within tBISTMode of the last bit of the EOP of the BIST message used to initiate the test is received by the Physical Layer.	PHYFSK_TX_LEV PHYFSK_TX_FREQ PHYFSK_TX_BIT PHYFSK_TX_EYE PHYFSK_TX_BER PHYFSK_TX_SPECT PHYFSK_RX_SQU PHYFSK_RX_BER	
6.5.8.1#2	For modes transmitting a continuous carrier signal (i.e. carrier modes and eye pattern) transmission shall start as soon as the UUT enters BIST mode.		
6.5.8.4 tBISTContMode			
6.5.8.4#1	"A UUT that has been put into a Continuous BIST Mode shall return to normal operation(either PE_SRC_Transition_to_default, PE_SNK_Transition_to_default, or PE_CBL_Ready) within tBISTContMode of the last bit of the bit of the EOP of GoodCRC Message sent in response to the BIST Message used to enable the Continuous BIST Mode."		
6.5.8.5 BISTReceiveErrorTimer			
6.5.8.5#1	The BISTReceiveErrorTimer shall be used by the sender's Protocol Layer during BIST operation to detect when a Test Frame has been lost and to trigger the transmission of the next test frame.	not tested	
6.5.8.5#2	The sender's Protocol Layer response when a BISTReceiveErrorTimer expires shall be to continue operation.	not tested	
6.5.8.5#3	The BISTReceiveErrorTimer shall be started when the nBISTPayload bit past the last bit of the SOP* has been transmitted by the Physical Layer.	not tested	
6.5.8.5#4	The BISTReceiveErrorTimer shall be stopped when the last bit of the EOP corresponding to the BIST message with a Data Object of Returned BIST Counters has been received by the Physical Layer.	not tested	
6.5.8.5#5	The Protocol Layer receiving a test frame shall respond with a BIST message with a Data Object of Returned BIST Counters within tTransmit in order to ensure that the sender's BISTReceiveErrorTimer does not expire.	not tested	
6.5.8.5#6	The tTransmit shall be measured from when the last bit of the test frame has been received by the Physical Layer until the first bit of the preamble of the BIST message has been transmitted by the Physical Layer.	not tested	
6.5.9.1 Swap Recovery Timer			
6.5.9.1#1	The Provider/Consumer shall wait tSwapRecover after either sending or receiving Hard Reset signaling before turning on its Power Source.	PROT_PROC_PSSOURCEOFFTIMER_SWPD_2 PROT_PROC_PSSOURCEONTIMER_2	
6.5.9.2 SwapSourceStartTimer			
6.5.9.2#1	The SwapSourceStartTimer shall be used by the new Source, after a Power Role Swap, to ensure that it does not send Source Capabilities Message before the new Sink is ready to receive the Source Capabilities Message.		
6.5.9.2#2	The new Source shall not send the Source Capabilities Message earlier than tSwapSourceStart after the last bit of the EOP of GoodCRC Message sent in response to the PS_RDY Message sent by the new Source indicating that its Power Supply is ready.	PROT_PROC_SWAP_TSTR_SRC_7	
6.5.9.2#3	The Sink shall be ready to receive a Source Capabilities Message tSwapSinkReady after having sent the last bit of the		

R2.0 Assertion #	Description	Test	Comments
	EOP of GoodCRC Message sent in response to the PS_RDY Message sent by the new Source indicating that its Power Supply is ready.		
6.5.10.1 HardResetCompleteTimer			
6.5.10.1#1	If the PHY Layer does not indicate that the Hard Reset signaling has been sent within tHardResetComplete of the Protocol Layer requesting transmission, then the Protocol Layer shall inform the Policy Engine that the Hard Reset signaling has been sent in order to ensure the Power Supply is reset in a timely fashion.		
6.5.10.2 PSHardResetTimer			
6.5.10.2#1	The Source shall wait tPSHardReset after sending Hard Reset signaling before starting to transition the VBUS voltage to vSafe0V.	PROT_PROC_HR_TSTR_1	
6.5.10.3 tDRSwapHardReset			
6.5.10.3#1	If a DR_Swap Message is received during Modal Operation then a Hard Reset shall be initiated by the recipient of the unexpected DR_Swap Message; Hard Reset Signaling shall be generated within tDRSwapHardReset of the EOP of the GoodCRC sent in response to the DR_Swap Message.		
6.5.11.1 VDMResponseTimer			
6.5.11.1#1	The VDMResponseTimer shall be used by the Initiator's Policy Engine to ensure that a Structured VDM Command request needing a response (e.g. Discover Identity Command request) is responded to within a bounded time of tVDMSenderResponse.		
6.5.11.1#2	The VDMResponseTimer shall be applied to all Structured VDM Commands except the Enter Mode and Exit Mode Commands which have their own timers (VDMModeEntryTimer and VDMModeExitTimer respectively).		
6.5.11.1#3	The Policy Engine's response when the VDMResponseTimer expires shall be dependent on the message sent (see Section 8.3).		
6.5.11.1#4	The VDMResponseTimer shall be started from the time the last bit of the GoodCRC message EOP (i.e. the GoodCRC message corresponding to the VDM Command requesting a response) has been received on VBUS by the Physical Layer.		
6.5.11.1#5	The VDMResponseTimer shall be stopped when the last bit of the expected VDM Command response EOP has been received by the Physical Layer.		
6.5.11.1#6	The receiver of a message requiring a response shall respond within tVDMReceiverResponse in order to ensure that the sender's VDMResponseTimer does not expire.		
6.5.11.1#7	The tVDMReceiverResponse time shall be measured from the time the last bit of the message EOP has been received by the Physical Layer until the first bit of the response message preamble has been transmitted on VBUS by the Physical Layer.		
6.5.11.2 VDMModeEntryTimer			
6.5.11.2#1	The VDMModeEntryTimer shall be used by the Initiator's Policy Engine to ensure that the response to a Structured VDM Enter Mode Command request (ACK, NAK or BUSY with ACK indicating that the requested Mode has been entered) arrives within a bounded time of tVDMWaitModeEntry.		
6.5.11.2#2	The VDMModeEntryTimer shall be started from the time the last bit of the GoodCRC message EOP (i.e. the GoodCRC message corresponding to the VDM Command request) has been received on VBUS by the Physical Layer.		
6.5.11.2#3	The VDMModeEntryTimer shall be stopped when the last bit		

R2.0 Assertion #	Description	Test	Comments
	of the expected Structured VDM Command response (ACK, NAK or BUSY) EOP has been received by the Physical Layer.		
6.5.11.2#4	The receiver of a message requiring a response shall respond within tVDMEnterMode in order to ensure that the sender's VDMModeEntryTimer does not expire.		
6.5.11.2#5	The tVDMEnterMode time shall be measured from the time the last bit of the message EOP has been received by the Physical Layer untel the first bit of the response message preamble has been transmitted on Vbus by the Physical Layer.		
6.5.11.3 VDMModeExitTimer			
6.5.11.3#1	The VDMModeExitTimer shall be used by the Initiator's Policy Engine to ensure that the ACK response to a Structured VDM Exit Mode Command, indicating that the requested Mode has been exited, arrives within a bounded time of tVDMWaitModeExit.		
6.5.11.3#2	The VDMModeExitTimer shall be started from the time the last bit of the GoodCRC message EOP (i.e. the GoodCRC message corresponding to the VDM Command requesting a response) has been received on VBUS by the Physical Layer.		
6.5.11.3#3	The VDMModeExitTimer shall be stopped when the last bit of the expected Structured VDM Command response ACK EOP has been received by the Physical Layer.		
6.5.11.3#4	The receiver of a message requiring a response shall respond within tVDMExitMode in order to ensure that the sender’s VDMModeExitTimer does not expire.		
6.5.11.3#5	The tVDMExitMode time shall be measured from the time the last bit of the message EOP has been received by the Physical Layer until the first bit of the response message preamble has been transmitted on VBUS by the Physical Layer.		
6.5.11.4 tVDMBusy			
6.5.11.4#1	The Initiator shall wait at least tVDMBusy, after receiving a BUSY Command response, before repeating the Structured VDM request again.		
6.5.12.1 VCONNOnTimer			
6.5.12.1#1	The VCONNOnTimer shall be started when: · The last bit of the EOP of the Accept message is received by the DFP. · The last bit of the EOP of GoodCRC message corresponding to the Accept message is received by the UFP.		
6.5.12.1#2	The VCONNOnTimer shall be stopped when: · The last bit of the EOP of the PS_RDY message is received by the DFP or UFP.		
6.5.12.1#3	The DFP or UFP, prior to sending the PS_RDY message, shall have turned VCONN On.		
6.5.12.2 VCONNOffTimer			
6.5.12.2#1	The initial VCONN source shall cease sourcing VCONN within tVCONNSourceOff of receipt of the last bit of the EOP of the PS_RDY message.		
6.5.13 tCableMessage			
6.5.13#1	The sender of an SOP’ or SOP’’ packet (either a DFP or Cable Plug), that is not a GoodCRC message, shall wait tCableMessage after the last bit of the EOP of the GoodCRC transmitted in response to the SOP’ or SOP’’ packet before sending another SOP’ or SOP’’ packet.	CAB_PROT_DISCOV_2 CAB_PROT_DISCOV_4 CAB_PROT_DISCOV_6 CAB_PROT_DISCOV_8	
6.5.13#2	Retransmission shall occur as described in Section 6.5.1 with tCableMessage applying to the last transmitted SOP’ or SOP’’ Packet.		
6.5.14 DiscoverIdentity Timer			

R2.0 Assertion #	Description	Test	Comments
6.5.14#1	When performing cable discovery during an Explicit Contract the Discover Identity Command request shall be sent every tDiscoverIdentity.		
6.5.14#2	No more than nDiscoverIdentityCount Discover Identity messages without a GoodCRC message response shall be sent.		
6.5.14#3	If no GoodCRC Message response is received after nDiscoverIdentityCount Discover Identity Command requests have been sent the DFP shall not send any further SOP'/SOP'' Messages.		
6.5.16 Time Values and Timers			
6.5.16#1	For each Timer Value, a given implementation shall pick a fixed value within the range specified. Table 6_31 lists the timers.		
6.6.1 MessageID Counter			
6.6.1#1	Each port shall maintain a copy of the last MessageID value received from its Port Partner.	A subject of the 'Test Jig for Multiple Port Testing'	
6.6.1#2	Devices that support multiple ports, such as Hubs, shall maintain copies of the last MessageID on a per port basis.	A subject of the 'Test Jig for Multiple Port Testing'	
6.6.1#3	A DFP or Source shich communicates using SOP* Packets shall maintain copies of the last MessageID for each type of SOP* it uses.	PROT_MSG_HDR_6	
6.6.1#4	The transmitter shall use the MessageID in a GoodCRC message to verify that a particular message was received correctly.	PROT_HDR	
6.6.1#5	The receiver shall use the MessageID to detect duplicate messages.	PROT_SEQ_GETSNKCAP_REJ	
6.6.1.1 Transmitter Usage			
6.6.1.1#1	The Transmitter shall use the MessageID as follows: · Upon receiving either Hard Reset signaling or a Soft Reset message, the transmitter shall set its MessageIDCounter to zero and re_initialize its retry mechanism.	PROT_HDR PROT_MSG_CTRL BMC_PHY_MSG_13	
6.6.1.1#2	The Transmitter shall use the MessageID as follows: · If a GoodCRC message with a MessageID matching the MessageIDCounter is not received before the CRCReceiveTimer expires, it shall retry the same packet up to nRetryCount times using the same MessageID.	PROT_HDR PROT_MSG_CTRL	
6.6.1.1#3	The Transmitter shall use the MessageID as follows: · If a GoodCRC message is received with a MessageID matching the current MessageIDCounter before the CRCReceiveTimer expires, the transmitter shall re_initialize its retry mechanism and increment its MessageIDCounter.	PROT_HDR PROT_MSG_CTRL	
6.6.1.1#4	"The Transmitter shall use the MessageID as follows: � If the message is aborted by the Policy Engine, the transmitter shall delete the message from its transmit buffer, re-initialize its retry mechanism and increment its MessageIDCounter."	not tested	
6.6.1.2 Receiver Usage			
6.6.1.2#1	The Receiver shall use the MessageID as follows: · When the first good packet is received after a reset, the receiver shall store a copy of the received MessageID value.	PROT_HDR_GCRC PROT_MSG_CTRL	
6.6.1.2#2	The Receiver shall use the MessageID as follows: · For subsequent messages, if MessageID value in a received message is the same as the stored value, the receiver shall return a GoodCRC message with that MessageID value and drop the message (this is a retry of an already received message). Note: this shall not apply to the Soft Reset message which always has a MessageID value of zero.	PROT_HDR_GCRC PROT_MSG_CTRL PROT_SEQ_GETSNKCAP_REJ	
6.6.1.2#3	The Receiver shall use the MessageID as follows: · If MessageID value in the received message is different	PROT_HDR_GCRC PROT_MSG_CTRL	

R2.0 Assertion #	Description	Test	Comments
	than the stored value, the receiver shall return a GoodCRC message with the new MessageID value, store a copy of the new MessageID value and process the message.		
6.6.2 Retry Counter			
6.6.2#1	If the nRetryCount retry fails, then the link shall be reset using the Soft Reset mechanism.		
6.6.2#2	Cable Plugs shall not retry messages.		
6.6.3 Hard Reset Counter			
6.6.3#1	Once the Hard Reset has been retried nHardResetCount times then it shall be assumed that the remote device is non_responsive.		
6.6.4 Capabilities Counter			
6.6.4#1	When the CapsCounter is implemented and the Source detects that a Sink is attached then after nCapsCount Source_Capabilities Messages have been sent the Source shall decide that the Sink is non-responsive, stop sending Source_Capabilities Messages and disable PD.		
6.6.4#2	A Sink shall use the SinkWaitCapTimer to trigger the resending of Source_Capabilities Messages by a USB Power Delivery capable Source which has previously stopped sending Source_Capabilities Messages.		
6.6.4#3	Any Sink which is attached and does not detect a Source_Capabilities Message, shall issue Hard Reset Signaling when the SinkWaitCapTimer times out in order to reset the Source.		
6.6.4#4	Resetting the Source shall also reset the CapsCounter and restart the sending of Source Capabilities messages.		
6.6.5 BIST Error Counter			
6.6.5#1	The Tester and UUT shall maintain a count of errors detected BISTErrorCounter (see Sections 6.4.3.1 and 6.4.3.2).	PHYFSK_RX_SQU PHYFSK_RX_BER	
6.6.5#2	The BISTErrorCounter shall contain the number of bits in error during a BIST Transmit or Receive test.	PHYFSK_RX_SQU PHYFSK_RX_BER	
6.6.5#3	The BISTErrorCounter shall be a 16_bit counter that shall be set to zero upon the BIST test start.	PHYFSK_RX_SQU PHYFSK_RX_BER	
6.6.5#4	The BISTErrorCounter shall freeze at a value of FFFFh.	not tested	
6.6.6 Discover Identity Counter			
6.6.6#1	When sending Discover Identity messages to a Cable Plug the DFP shall maintain a count of messages sent (DiscoverIdentityCounter).		
6.6.6#2	No more than nDiscoverIdentityCount Discover Identity messages shall be sent by the DFP receiving a GoodCRC message response.		
6.6.6#3	A Data Role Swap shall reset the DiscoverIdentityCounter to zero.		
6.6.7 VDMBusy Counter			
6.6.7#1	When sending Responder Busy responses to a Structured Vendor Defined Message a UFP or Cable Plug shall maintain a count of messages sent (VDMBusyCounter).	CAB_PROT_DISCOV BMC_PROT_DISCOV	
6.6.7#2	No more than nBusyCount Responder Busy responses shall be sent.	CAB_PROT_DISCOV BMC_PROT_DISCOV	
6.6.7#3	The VDMBusyCounter shall be reset on sending a non_Busy response.		
6.7.1 Soft Reset			
6.7.1#1	It shall not have any impact on power supply operation, but shall be used whenever there is a protocol error.		
6.7.1#2	Protocol Errors that shall lead to a Soft Reset are unexpected messages during one of the atomic message sequences defined in Section 8.3.2.		

R2.0 Assertion #	Description	Test	Comments
6.7.1#3	An unrecognized message, received in the PE_SNK_Ready or PE_SRC_Ready states, shall not cause a Soft Reset message to be generated but shall be ignored.		
6.7.1#4	This shall cause the Source to send a Soft Reset message, transmission of which shall be completed within tSoftReset of the CRCReceiveTimer expiring.		
6.7.1#5	"A Soft Reset shall impact the USB Power Delivery layers in the following ways: � Physical Layer: Reset not required since the Physical Layer resets on each packet transmission/reception. � Protocol Layer: Reset MessageIDCounter, RetryCounter and state machines. � Policy Engine: Reset state dependent behavior by performing an Explicit Contract Negotiation. � Power Supply: Shall not change."		
6.7.1#6	Message numbers shall be set to zero prior to sending the Soft Reset/Accept message since the issue may be with the counters.		
6.7.1#7	The sender of a Soft reset message shall reset its MessageIDCounter and RetryCounter, the receiver of the message shall reset its MessageIDCounter and RetryCounter before sending the Accept message response.		
6.7.2.1 Hard Reset Common Requirements			
6.7.2.1#1	Both the sender and recipient shall cause their Power Supplies to return to their default states (see Sections 7.3.12 to 7.3.17 for details of voltage transitions).		
6.7.2.1#2	In addition their respective Protocol Layers shall be reset as for the Soft Reset.		
6.7.2.2 Type_A/B and Hard Reset			
6.7.2.2#1	When there has been a Type_A/B Power Role Swap, a Hard Reset shall cause the Port Partners to return to their default Source/Sink roles.		
6.7.2.2#2	A Type_A Port shall return to Source operation.		
6.7.2.2#3	A Type_B Port shall return to Sink operation.		
6.7.2.3 Type_C and Hard Reset			
6.7.2.3#1	A Hard Reset shall not cause any change to either the Rp/Rd resistor being asserted, to the Data Role or to the Source of VCONN.		
6.7.2.3#2	If there has been a Data Role Swap the Hard Reset shall cause the Port Data Role to be changed back to DFP for a Port with the Rp resistor asserted and UFP for a Port with the Rd resistor asserted.		
6.7.2.3#3	When VCONN is supported (see [USBType-C 1.0]) the Hard Reset shall cause the Port with the Rp resistor asserted to supply VCONN and the Port with the Rd resistor asserted to turn off VCONN.		
6.7.2.4 Cable Plugs and Hard Reset			
6.7.2.4#1	Cable Plugs shall not generate Hard Reset signaling but shall monitor for Hard Reset signaling between the Port Partners and shall reset when this is detected (see Section 8.3.3.8.3).		
6.7.2.4#2	The Cable Plugs shall perform the equivalent of a power cycle returning to their initial power up state.		
6.7.2.5 Modal Operation and Hard Reset			
6.7.2.5#1	A Hard Reset shall cause all Active Modes to be exited by both Port Partners and any Cable Plugs (see Section 6.4.4.3.4).		
6.7.3 Cable Reset			
6.7.3#1	Cable Resets are signaled by an ordered set as defined in Section 5.6.5.	CAB_PHY_MSG BMC_PHY_MSG	

R2.0 Assertion #	Description	Test	Comments
6.7.3#2	Both the sender and recipient of Cable Reset signaling shall reset their respective Protocol Layers.	CAB_PHY_MSG BMC_PHY_MSG	
6.7.3#3	The Cable Plugs shall perform the equivalent of a power cycle returning to their initial power up state.	CAB_PHY_MSG BMC_PHY_MSG	
6.7.3#5	If VCONN has been turned off the DFP shall turn on VCONN prior to generating Cable Reset signaling.		
6.7.3#6	If there has been a VCONN Swap and the UFP is currently supplying VCONN, the DFP shall perform a VCONN Swap such that it is supplying VCONN prior to generating Cable Reset signaling.		
6.7.3#7	Only a DFP shall generate Cable Reset signaling.		
6.7.3#8	A DFP shall only generate Cable Reset signaling within an Explicit Contract.		
6.7.3#9	A Cable Reset shall cause all Modes currently active in the Cable Plugs to be exited (see Section 6.4.4.3.4).	CAB_PHY_MSG_14	
6.8.1 Introduction to state diagrams used in Chapter 6			
6.8.1#1	The state diagrams defined in Section 6.8 are normative and shall define the operation of the Power Delivery protocol layer.		
6.8.2 State Operation			
6.8.2#1	For each SOP* Communication being sent and received there shall be separate Protocol Layer Transmission and Protocol Layer Reception and BIST State Machine instances, with their own counter and timer instances.		
6.8.2#2	Soft Reset shall only apply to the State Machine instances it is targeted at based on the type of SOP* Packet used to send the Soft Reset message.		
6.8.2#3	The Hard Reset State Machine (including Cable Reset) shall apply simultaneously to all Protocol Layer State Machine instances active in the DFP, UFP and Cable Plug (if present).		
6.8.2.1.1 PRL Tx PHY Layer Reset state			
6.8.2.1.1#1	The Protocol Layer shall enter the PRL_Tx_PHY_Layer_Reset state: · At startup. · As a result of a Soft Reset request being received by the PHY Layer. · On exit from a Hard Reset.		
6.8.2.1.1#2	On entry to the PRL_Tx_PHY_Layer_Reset state the the Protocol Layer shall reset the PHY Layer (clear any outstanding messages and enable communications).		
6.8.2.1.1#3	The Protocol Layer shall transition to the PRL_Tx_Wait_for_Message_Request state when: · When the PHY Layer reset is complete.		
6.8.2.1.2 PRL Tx Wait for Message Request state			
6.8.2.1.2#1	This shall be the startup state for the Protocol Layer message transmission and also the starting state when exiting from a Hard Reset.		
6.8.2.1.2#2	On entry to the PRL Tx Wait for Message Request state the Protocol Layer shall reset the PHY and shall then reset the RetryCounter.		
6.8.2.1.2#3	The Protocol Layer shall transition to the PRL Tx Construct message state when: · A message request is received from the Policy Engine which is not a Soft Reset message.		
6.8.2.1.2#4	The Protocol Layer shall transition to the PRL Tx Layer Reset for Transmit state when: · A message request is received from the Policy Engine which is a Soft Reset message.		

R2.0 Assertion #	Description	Test	Comments
6.8.2.1.3 PRL Tx Layer Reset for Transmit state			
6.8.2.1.3#1	On entry to the PRL Tx Layer Reset for Transmit state the Protocol Layer shall reset the MessageIDCounter.		
6.8.2.1.3#2	The Protocol Layer shall transition Protocol Layer message reception to the PRL Rx Wait for first PHY Message state (See Section 6.8.2.2.1) in order to reset the stored MessageID.		
6.8.2.1.3#3	The Protocol Layer shall transition to the PRL Tx Construct Message State when: · The layer Reset actions in this state have been completed.		
6.8.2.1.4 PRL Tx Construct Message State			
6.8.2.1.4#1	On entry to the PRL Tx Construct Message state the Protocol Layer shall construct the message requested by the Policy Engine, or resend a previously constructed message, and then pass this message to the PHY Layer.		
6.8.2.1.4#2	"The Protocol Layer shall transition to the Wait for PHY Response state when: The message has been sent to the PHY Layer."		
6.8.2.1.5 PRL Tx Wait for PHY Response State			
6.8.2.1.5#1	On entry to the PRL Tx Wait for PHY Response State, once the message has been sent, the Protocol Layer shall initialize and run the CRCReceiveTimer (see Section 6.5.1).		
6.8.2.1.5#2	The Protocol Layer shall transition to the PRL Tx Match MessageID state when: · A GoodCRC message response is receive from the PHY Layer.		
6.8.2.1.5#3	The Protocol Layer shall transition to the PRL Tx Check RetryCounter state when: • The CRCReceiveTimer times out. • Or the PHY Layer indicates that a message has been discarded due to the channel being busy but the channel is now idle (see Section 5.8).		
6.8.2.1.6 PRL Tx Match MessageID State			
6.8.2.1.6#1	On entry to the PRL Tx Match MessageID state the Protocol Layer shall compare the MessageIDCounter and the MessageID of the received GoodCRC message.	PROT_MSG_HDR_GCRC_3	
6.8.2.1.6#2	The Protocol Layer shall transition to thePRL Tx Message Sent state when: · The MessageIDCounter and the MessageID of the received GoodCRC message match.		
6.8.2.1.6#3	The Protocol Layer shall transition to the PRL Tx Check RetryCounter state when: · The MessageIDCounter and the MessageID of the received GoodCRC message do not match.		
6.8.2.1.7 PRL Tx Message Sent state			
6.8.2.1.7#1	On entry to the PRL Tx Message Sent state the Protocol Layer shall increment the MessageIDCounter and inform the Policy Engine that the message has been sent.		
6.8.2.1.7#2	The Protocol Layer shall transition to the PRL Tx Wait for Message Request state when: · The Policy Engine has been informed that the message has been sent.		
6.8.2.1.8 PRL Tx Check RetryCounter State			
6.8.2.1.8#1	On entry to the PRL Tx Check RetryCounter state the Protocol Layer in a DFP or UFP shall increment the value of the RetryCounter and then check it in order to determine whether it is necessary to retry sending the message.		
6.8.2.1.8#2	The Protocol Layer shall transition to the PRL Tx Construct Message state in order to retry message sending when: · RetryCounter ≤ nRetryCount, and		

R2.0 Assertion #	Description	Test	Comments
	_ This is not a Cable Plug.		
6.8.2.1.8#3	The Protocol Layer shall transition to the PRL Tx Transmission Error state when: · RetryCounter > nRetryCount, or _ This is a Cable Plug which does not retry.		
6.8.2.1.9 PRL Tx Transmission Error state			
6.8.2.1.9#1	On entry to the PRL Tx Transmission Error state the Protocol Layer shall increment the MessageIDCounter and inform the Policy Engine of the transmission error.		
6.8.2.1.9#2	The Protocol Layer shall transition to the PRL Tx Wait for Message Request state when: · The Policy Engine has been informed of the transmission error.		
6.8.2.1.10 PRL Tx Discard Message state			
6.8.2.1.10#1	Protocol Layer message transmission shall enter the PRL Tx Discard Message state whenever Protocol Layer Message reception receives an incoming message.		
6.8.2.1.10#2	On entry to the the PRL Tx Discard Message state, if there is a message queued awaiting transmission, the Protocol Layer shall discard the message and increment the MessageIDCounter.		
6.8.2.1.10#3	The Protocol Layer shall transition to the PRL Tx PHY Layer Reset state when: • Discarding is complete i.e. the message queue is empty.		
6.8.2.2.1 PRL Rx Wait for PHY Message state			
6.8.2.2.1#1	The Protocol Layer shall enter the PRL_Rx_Wait_for_PHY_Message state: · At startup. · As a result of a Soft Reset request from the Policy Engine. · On exit from a Hard Reset.		
6.8.2.2.1#2	The Protocol Layer shall transition to the PRL_Rx_Send_GoodCRC state when: · A message is passed up from the PHY Layer.		
6.8.2.2.1#3	The Protocol Layer shall transition to the PRL_Rx_Layer_Reset_for_Receive state when: · A Soft Reset message is received from the PHY Layer.		
6.8.2.2.2 PRL Rx Layer Reset for Receive State			
6.8.2.2.2#1	On entry to the PRL RX Layer Reset for Receive state the Protocol Layer shall reset the MessageIDCounter and clear the stored MessageID.		
6.8.2.2.2#2	The Protocol Layer shall transition Protocol Layer message transmission to the PRL Tx Wait for Message Request state (See Section 6.8.2.1.1).		
6.8.2.2.2#3	The Protocol Layer shall transition to the Send PRL Rx Send GoodCRC State when: · The Soft Reset actions in this state have been completed.		
6.8.2.2.3 PRL Rx Send GoodCRC state			
6.8.2.2.3#1	On entry to the PRL Rx Send GoodCRC state the Protocol Layer shall construct a GoodCRC message and request the PHY Layer to transmit it.		
6.8.2.2.3#2	"The Protocol Layer shall transition to the PRL Rx Check MessageID state when: The GoodCRC message has been passed to the PHY Layer"		
6.8.2.2.3#3	"When the PHY Layer indicates that a message has been discarded due to VBUS or CC being busy but VBUS or CC is now idle (see Section 5.7), the Protocol Layer shall either: Transition to the PRL_Rx_Check_MessageID state or Transition to the PRL_Rx_Wait_for_PHY_Message state"		
6.8.2.2.4 PRL Rx Check MessageID state			

R2.0 Assertion #	Description	Test	Comments
6.8.2.2.4#1	On entry to the PRL Rx Check MessageID state the Protocol Layer shall compare the MessageID of the received message with its stored value if a value has been previously stored.		
6.8.2.2.4#2	The Protocol Layer shall transition to the PRL Rx Wait For PHY Message state when: · The MessageID of the received message equals the stored MessageID value since this is a message retry which shall be discarded.		
6.8.2.2.4#3	The Protocol Layer shall transition to the PRL Rx Store MessageID state when: · The MessageID of the received message does not equal the stored MessageID value since this is a new message or · This is the first received message and no MessageID value is currently stored.		
6.8.2.2.5 PRL Rx Store MessageID			
6.8.2.2.5#1	On entry to the PRL Rx Store MessageID state the Protocol Layer shall transition Protocol Layer message transmission to the PRL Tx Discard Message state (except when a Ping message has been received in which case the PRL Tx Discard Message state should not be entered), replace the stored value of MessageID with the value of MessageID in the received message (not required for Cable Plugs) and pass the message up to the Policy Engine.		
6.8.2.2.5#2	The Protocol Layer shall transition to the PRL Rx Wait for PHY Message state when: · The message has been passed up to the Policy Engine.		
6.8.2.3.1 PRL HR Reset Layer State			
6.8.2.3.1#1	"The Protocol Layer shall enter the PRL HR Reset Layer state from any other state when: | A Hard Reset Request is received from the Policy Engine or Hard Reset signaling is received from the Physical Layer or A Cable Reset Request is received from the Policy Engine or Cable Reset signaling is received from the Physical Layer"		
6.8.2.3.1#2	On Entry to the PRL HR Reset Layer state the Protocol Layer shall reset the MessageIDCounter.	BMC_PHY_MSG_14	
6.8.2.3.1#3	It shall also reset the states of the Protocol Layer transmission and reception state machines to their starting points.		
6.8.2.3.1#4	The Protocol Layer transmission state machine shall transition to the PRL Tx Wait for Message Request state.		
6.8.2.3.1#5	The Protocol Layer reception state machine shall transition to the PRL Rx Wait For First PHY Message state.		
6.8.2.3.1#6	"The Protocol Layer shall transition to the PRL_HR_Request_Hard_Reset state when: The Protocol Layer's reset is complete and o The Hard Reset request has originated from the Policy Engine or o The Cable Reset request has originated from the Policy Engine."		
6.8.2.3.1#7	The Protocol Layer shall transition to the PRL HR Indicate Hard Reset state when: · The Protocol Layer's reset is complete and · The Hard Reset request has been passed up from the Physical Layer or · A Cable Reset request has been passed up from the Physical Layer (Cable Plug only).		
6.8.2.3.2 PRL HR Indicate Hard Reset state			
6.8.2.3.2#1	On entry to the PRL_HR_Indicate_Hard_Reset state the		

R2.0 Assertion #	Description	Test	Comments
	Protocol Layer shall indicate to the Policy Engine that either Hard Reset signaling or Cable Reset signaling has been received.		
6.8.2.3.2#2	The Protocol Layer shall transition to the PRL_HR_Wait_for_PE_Hard_Reset_Complete state when: · The Indication to the Policy Engine has been sent.		
6.8.2.3.3 PRL HR Request Hard Reset state			
6.8.2.3.3#1	On entry to the PRL_HR_Request_Hard_Reset state the Protocol Layer shall request the Physical Layer to send either Hard Reset signaling or Cable Reset signaling .		
6.8.2.3.3#2	"The Protocol Layer shall transition to the PRL_HR_Wait_for_PHY_Hard_Reset_Complete state when: ? The Physical Layer Hard Reset signaling request has been sent or ? The Physical Layer Cable Reset signaling request has been sent."		
6.8.2.3.4 PRL HR Wait for PHY Hard Reset Complete			
6.8.2.3.4#1	In the PRL_HR_Wait_for_PHY_Hard_Reset_Complete state the Protocol Layer shall start the HardResetCompleteTimer and wait for the PHY Layer to indicate that the Hard Reset or Cable Reset has been completed.		
6.8.2.3.4#2	"The Protocol Layer shall transition to the PRL_HR_PE_Hard_Reset_1 Complete state when: ? A Hard Reset complete indication is received from the PHY Layer or ? A Cable Reset indication is received from the PHY Layer or ? The HardResetCompleteTimer times out."		
6.8.2.3.5 PRL HR PHY Hard Reset Requested			
6.8.2.3.5#1	On entry to the PRL_HR_PHY_Hard_Reset_Requested state the Protocol Layer shall inform the Policy Engine that the PHY Layer has been requested to perform a Hard Reset or Cable Reset.		
6.8.2.3.5#2	The Protocol Layer shall transition to thePRL HR Wait for PE Hard Reset Complete state when: • The Indication to the Policy Engine has been sent.		
6.8.2.3.6 PRL HR Wait for PE Hard Reset Complete			
6.8.2.3.6#1	In the PRL_HR_Wait_for_PE_Hard_Reset_Complete state the Protocol Layer shall wait for the Policy Engine to indicate that the Hard Reset or Cable Reset has been completed.		
6.8.2.3.6#2	"The Protocol Layer shall transition to the PRL_HR_PE_Hard_Reset_Complete state when: ? A Hard Reset complete indication is received from the Policy Engine or ? A Cable Reset complete indication is received from the Policy Engine."		
6.8.2.3.7 PRL HR PE Hard Reset Complete			
6.8.2.3.7#1	On entry to the PRL_HR_PE_Hard_Reset_Complete state the Protocol Layer shall inform the Physical Layer that the Hard Reset or Cable Reset is complete.		
6.8.3.7#2	"The Protocol Layer shall exit from the Hard Reset and return to normal operation when: ? The Physical Layer has been informed that the Hard Reset is complete so that it will re_enable the communications channel. If Hard Reset signaling is still pending due to a non_idle channel this shall be cleared and not sent or ? The Physical Layer has been informed that the Cable Reset is complete."		
6.8.3.1.1 PRL BIST Tx Init PRBS			
6.8.3.1.1#1	The Protocol Layer shall enter the PRL BIST Tx Init PRBS state		

R2.0 Assertion #	Description	Test	Comments
	from the PRL Tx Wait For Message Request state (see Section 6.8.2.1.1) when: • The Policy Engine requests the Protocol Layer to enter BIST Transmitter Test Mode.		
6.8.3.1.1#2	On entry to thePRL BIST Tx Init PRBS state the Protocol Layer shall request the Physical Layer to reset the BISTErrorCounter and preload the PRBS generator.		
6.8.3.1.1#3	The Protocol Layer shall transition to the PRL BIST Tx Wait Test Frame Request state when: • The PRBS generator has been preloaded.		
6.8.3.1.2 PRL BIST Tx Wait Test Frame State			
6.8.3.1.2#1	On entry to the PRL BIST Tx Wait Test Frame Request state the Protocol Layer shall wait for a test frame from the Policy Engine.		
6.8.3.1.2#2	The Protocol Layer shall transition to the PRL BIST Tx Test Frame state when: • When a request to transmit a test frame is received from the Policy Engine.		
6.8.3.1.3 PRL BIST Tx Test Frame			
6.8.3.1.3#1	On entry to the PRL BIST Tx Test Frame state the Protocol Layer shall request the Physical Layer to construct the Test Frame.		
6.8.3.1.3#2	The Protocol Layer shall transition to the PRL BIST Tx PHY Response state when: · The Test Frame request has been sent to the Physical Layer.		
6.8.3.1.2 PRL BIST Tx PHY Response state			
6.8.3.1.4#1	In the PRL BIST Tx PHY Response state the Protocol Layer shall wait for the Physical Layer to provide a response to the Test Frame.		
6.8.3.1.4#2	On entry to the PRL BIST Tx PHY Response state the Protocol Layer shall initialize and run the BISTReceiveErrorTimer.		
6.8.3.1.4#3	On exit from the PRL BIST Tx PHY Response state the Protocol Layer shall inform the Policy Engine either of the received Error Counter value or of a BISTReceiveErrorTimer timeout.		
6.8.3.1.4#4	The Protocol Layer shall transition to the PRL BIST Tx Wait Test Frame Request (see Section 6.8.2.1.5) when: · An error counter response is received from the Physical Layer. • Or the BISTReceiveErrorTimer times out.		
6.8.3.2.1 PRL BIST Rx Reset Counter state			
6.8.3.2.1#1	The Protocol Layer shall enter the PRL BIST Rx Reset Counter state from the PRL Rx Wait For Message From PHY state when: · A BIST Receiver Test Mode request is received from the Policy Engine.		
6.8.3.2.1#2	On entry to the PRL BIST Rx Reset Counter state the Protocol Layer shall request the Physical Layer to reset the BISTErrorCounter.		
6.8.3.2.1#3	The Protocol Layer shall transition to the PRL BIST Rx Test Frame state when: · The BISTErrorCounter has been reset by the Physical Layer.		
6.8.3.2.2 PRL BIST Rx Test Frame state			
6.8.3.2.2#1	In the PRL BIST Rx Test Frame State the Protocol Layer shall wait for the Physical Layer to receive the next test frame.		
6.8.3.2.2#2	The Protocol Layer shall transition to the PRL BIST Rx Error Count state when: · The Physical Layer has received the next Test Frame and passes up the current value of the BISTErrorCounter is received from the PHY Layer.		

R2.0 Assertion #	Description	Test	Comments
6.8.3.2.3 PRL BIST Rx Error Count State			
6.8.3.2.3#1	On entry to the PRL BIST Rx Error Count state the Protocol Layer shall construct a BIST message with a Data Object of Returned BIST Counters using the BISTErrorCounter value returned by the Physical Layer.		
6.8.3.2.3#2	This BIST message shall be passed to the Physical Layer for transmission.		
6.8.3.2.3#3	The Protocol Layer shall transition to the PRL BIST Rx Inform Policy state when: · The BIST message has been sent.		
6.8.3.2.4 PRL BIST Rx Inform Policy state			
6.8.3.2.4#1	On entry to the PRL BIST Rx Inform Policy state the Protocol Layer shall inform the Policy Engine that the BIST message containing the BISTErrorCounter has been sent.		
6.8.3.2.4#2	The Protocol Layer shall transition to the PRL BIST Rx Test Frame state when: · The Policy Engine has been informed that the BIST message containing the BISTErrorCounter has been sent.		
6.9 Message Applicability			
6.9#1	When a Message is supported the feature and Message sequence implied by the Message shall also be supported.		
6.9#2	For example Sinks using power for charging that support the GotoMin Message shall be able to reduce their current draw when requested via a GotoMin Message.	PROT_PROC_GOTOMIN_TSTR_1	
6.9#3	The following abbreviations are used: · M – Mandatory; shall be supported by this Port/Cable Plug · CM – Conditionally Mandatory; shall be supported by a given Port/Cable Plug based on features · R – Recommended; should be supported by this Port/Cable Plug · O – Optional; may be supported by this Port/Cable Plug · NS – Not Supported; shall not by transmitted by this Port/Cable Plug and shall be Ignored by this Port/Cable Plug when received. · RJ – Reject; this Port/Cable Plug shall return a Reject Message when received · NK – NAK; this Port/Cable Plug shall return Responder NAK to this Command when received		
6.9#4	Note: that where NS/RJ/NK is indicated for Received Messages this shall apply to the PE_CBL_Ready, PE_SNK_Ready or PE_SRC_Ready states only since unexpected Messages received during a Message sequence are Protocol Errors (see Section 6.7.1).		
6.9#5	Table 6-35 details Control Messages (except for those specific to the Type-C connector) that shall/1 should/shall not be transmitted and received by a Source, Sink or Cable Plug.		
6.9#6	Requirements for Dual-Role Power Ports shall override any requirements for Source-only or Sink-Only Ports.		
6.9#7	(Table 6-35) Note 1: Shall be supported by a Hub with multiple Downstream Ports. Should be supported by a Host with multiple Downstream Ports.		
6.9#8	(Table 6-35) Note 2: Shall be supported by BFSK systems.		
6.9#9	(Table 6-35) Note 3: Shall be supported when transmision of GotoMin Messages is supported.		
6.9#0	Table 6-36 details Control Messages specific to the Type-C connector that shall/should/shall not be transmitted and received by a DFP, UFP or Cable Plug.		
6.9#11	Requirements for Dual-Role Data Ports shall override any requirements for DFP-only or UFP-Only Ports.		

R2.0 Assertion #	Description	Test	Comments
6.9#12	(Table 6-36) Note 1: Shall be supported by any Type-C Port that utilizes the SSTX and SSRX pins and supports PR_Swap Message		
6.9#13	Table 6-37 details Data Messages (except for VDM Commands) that shall/should/shall not be transmitted and received by a Source, Sink or Cable Plug.		
6.9#14	Requirements for Dual-Role Power Ports shall override any requirements for Source-only or Sink-Only Ports.		
6.9#15	(Table 6-37) Note 2: For details of which BIST Modes and Messages shall be supported see Sections 5.9.9 and 6.4.3.		
6.9#16	Table 6-38 details VDM Commands that shall/should/shall not be transmitted and received by a DFP, UFP or Cable Plug.		
6.9#17	Requirements for Modal Operation shall override any requirements for DFP or UFP Ports or Cable Plugs without Modal Operation.		

7.5.2 Recommended (Should)

R2.0 Assertion #	Description	Test	Comments
Subsection 6.3 Control Message			
Subsection 6.3.12.2 Wait in response to a PR_Swap Message			
6.3.12.2#r1	Once it has completed this evalution one of the Port Partners should initiate the Power Role swap process again by sending a PR_Swap message.		
6.3.12.2#r2	Once the System Policy Manager has completed its evaluation of the request the hub should initiate the Power Role swap process again by sending a PR_Swap message.		
6.3.12.3 Wait in response to a DR_Swap Message			
6.3.12.3#r1	Once it has completed this evalution one of the Port Partners should initiate the Data Role swap process again by sending a DR_Swap message.		
6.3.12.4 Wait in response to a VCONN_Swap Message			
6.3.12.4#r1	DFP should initiate the VCONN Swap process again at a future time by resending a VCONN_Swap message.		
6.4.2.5 No USB Suspend			
6.4.2.5#r1	The Source uses this flag to evaluate whether it should re_issue the Source capabilities message with the USB Suspend flag cleared.		
6.4.4.3 Use of Commands			
6.4.4.3#r1	A NAK response should be taken as an indication not to retry that particular Command.		
6.4.4.3.1.8 Product VDO			
6.4.4.3.1.8#r1	Manufacturers should set the USB Product ID field to a unique value identifying the product and should set the bcdDevice field to a version number relevant to the release version of the product.		
6.8.2.2 Protocol Layer Message Reception			
6.8.2.2#r1	In the case of a Ping message being received, in order to maintain robust communications in the presence of collisions, the outgoing message should not be discarded.		

7.6 Chapter 7 Power Supply

7.6.1 Normative (Shall/Normative)

R2.0 Assertion #	Description	Test	Comments
Subsection 7.1 Source Requirements			
Subsection 7.1.1 Behavioral Aspects			
7.1.1#1	The Source in the USB Power Delivery system shall exhibit the following behaviors: _Shall be backward compatible with legacy VBUS ports.	[USB2.0], [USB3.1] and [BC1.2] testing	
7.1.1#2	"The Source in the USB Power Delivery system shall exhibit the following behaviors. Shall supply the default [USB2.0], [USB3.1], [USBType_C1.0] or [BC1.2] voltage and current to VBUS when the USB cable is attached (USB Default Operation)."		
7.1.1#3	"The Source in the USB Power Delivery system shall exhibit the following behaviors. Shall supply the default [USB2.0], [USB3.1], [USBType_C1.0] or [BC1.2] voltage and current to VBUS when a USB Power Delivery contract does not exist (USB Default Operation)."	[USB2.0], [USB3.1] and [BC1.2] testing	
7.1.1#4	The Source in the USB Power Delivery system shall exhibit the following behaviors. · Shall return vSafe0V for some time then return to vSafe5V when a Hard Reset signaling is received.	PROT_PROC_PSSOURCEOFFTIMER_3 PROT_PROC_PSSOURCEOFFTIMER_SWPD_3 PROT_PROC_PSSOURCEONTIMER_3 PROT_PROC_PSSOURCEONTIMER_SWPD_3	
7.1.1#5	The Source in the USB Power Delivery system shall exhibit the following behaviors. · Shall control VBUS voltage transitions as bound by undershoot, overshoot and transition time requirements.	POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.1#6	The Source in a Consumer/Provider exhibits the following behaviors: • Shall support Dead Battery operation as defined in Section 4.1.1 for Type_A to Type_B connections and as defined in [USBType_C1.0] for Type_C to Type_C connections.	POW_DB_CP	
7.1.1#7	The Source in a Consumer/Provider exhibits the following behaviors: • Shall return to Sink operation when Hard Reset signaling is received		
7.1.1#8	The Source in a Consumer/Provider exhibits the following behaviors: • Shall control VBUS voltage transitions as bound by undershoot, overshoot and transition time requirements	POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.2 Source Bulk Capacitance			
7.1.2#1	The Source shall have a bulk capacitance located between the output of the power supply and the transceiver isolation impedance as shown in Figure 7_1.	Vendor Declaration	
7.1.2#2	The Source bulk capacitance shall not be placed between the transceiver isolation impedance and the USB receptacle.	PHYFSK_TX_LEV PHYFSK_RX_BER	
7.1.2#3	The capacitance change shall occur before the Source is ready to operate at the new power level.	Vendor Declaration	
7.1.2#4	During a Power Role Swap, the Default Source shall transition to Swap Standby before operating as the new Sink.	PROT_PROC_SWAP_TSTR_SRC_8 PROT_PROC_SWAP_UUT_SNK_3	
7.1.2#5	Any change in bulk capacitance required to complete the Power Role Swap shall occur during Swap Standby.	Vendor Declaration	
7.1.3 Types of Sources			
7.1.3#1	Sources shall support at least one fixed power source capable of supplying vSafe5V.	PROT_MSG_DATA_SRC_CAP POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.3#2	The output voltage of a Fixed Supply shall be specified in terms of an absolute tolerance, vSrcNew, relative to the	PROT_MSG_DATA_SRC_CAP POW_SRC_LOAD_P_PC	

R2.0 Assertion #	Description	Test	Comments
	nominal value.	POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.3#3	The output voltage of a Variable Power Supply (non_battery) shall be specified as vSrcNew in terms of an absolute maximum output voltage and an absolute minimum output voltage.	PROT_MSG_DATA_SRC_CAP POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.3#4	The output voltage of a Battery supply shall be specified as vSrcNew in terms of an absolute maximum output voltage and an absolute minimum output voltage.	PROT_MSG_DATA_SRC_CAP POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.4 Positive Voltage Transitions			
7.1.4#1	The Source shall transition VBUS from the starting voltage to the higher new voltage in a controlled manner.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.4#2	The negotiated new voltage (e.g., 12V or 20V) defines the typical value for vSrcNew. If the newly negotiated voltage is 5V, then vSafe5V limits shall apply.		
7.1.4#3	During the positive transition the Source shall be able to supply the Sink standby power and the transient current to charge the total bulk capacitance on VBUS.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.4#4	The slew rate of positive transition shall not exceed vSrcSlewPos.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.4#5	The Source output voltage during positive transitions shall settle within the Source output tolerance range vSrcNew.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.4#6	The Source shall be able to supply the newly negotiated power level at the new voltage by tSrcReady.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.4#7	The positive voltage transition shall remain monotonic while the transitioning voltage is below vSrcValid min and shall remain within the vSrcValid range upon crossing vSrcValid min as shown in Figure 7_2.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.4#8	When the starting voltage on VBUS for the transition is vSafe5V, the VBUS voltage shall not droop below vSafe5V min.		
7.1.4#9	When the starting voltage on VBUS is any voltage other than vSafe5V, the VBUS voltage shall not droop below vSrcValid min.		
7.1.4#10	During normal operation, not including VBUS transitions or discharge, VBUS shall not go beyond the limits of vSrcValid (or vSafe5V if VBUS is 5V).		
7.1.4#11	This limitation shall apply to static and transient VBUS behavior across all single port and multi_port power configurations.		
7.1.5 Negative Voltage Transitions			
7.1.5#1	If the newly negotiated voltage is 5V, then vSafe5V limits shall apply.		
7.1.5#2	The slew rate of the negative transition shall not exceed vSlewRateNeg.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.5#3	The negative voltage transistion shall remain monotonic while the transitioning voltage is above vSrcValid max.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.5#4	The negative voltage transition … and shall remain within the vSrcValid range upon crossing vSrcValid max as shown in Figure 7_3.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.6 Response to Hard Resets			

R2.0 Assertion #	Description	Test	Comments
7.1.6#1	Hard Reset signaling indicates a communication failure has occurred and the Source shall drive VBUS to vSafe0V as shown in Figure 7_4.	[PROT_PROC_HR_TSTR_3]	
7.1.6#2	After establishing the safe voltage condition on VBUS, the power supply shall wait tSrcRecover before powering VBUS to vSafe5V.	[PROT_PROC_HR_TSTR_4]	
7.1.6#3	A source using a Type-C connector shall conform to the Vconn timing as specified in [USBType-C 1.0]		
7.1.6#4	"When a Hard Reset occurs the Source shall start to transition the VBUS voltage to vSafe0V and the VCONN voltage to vVConnDischarge (see [USBType-C 1.0]) either: tPSHardReset after the last bit of the Hard Reset Signaling has been received from the Sink or tPSHardReset after the last bit of the Hard Reset Signaling has been sent by the Source"	[PROT_PROC_HR_TSTR_1]	
7.1.6#5	The Source shall meet both tSafe5V and tSafe0V relative to the start of the voltage transition as shown in Figure 7_4	[PROT_PROC_HR_TSTR_1] [PROT_PROC_HR_TSTR_2] [PROT_PROC_HR_TSTR_3]	
7.1.6#6	The Source shall meet tVConnDischarge relative to the start of the voltage transition as shown in Figure 7-5.		
7.1.6#7	The source shall meet tVconnOn relative to VBUS reaching vSafe5V.		
7.1.7 Changing the Output Power Capability			
7.1.7#1	The Source shall be able to supply a higher or lower load current within tSrcReady.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.8 Safe Operating Considerations			
7.1.8#1	The Source shall provide short circuit current limiting to protect its port from prolonged exposure to excessive current draw.	Vendor declaration VEND_GEN	
7.1.8#2	The maximum steady state operating current of the Source shall be the lesser of these two values.	Vendor declaration	
7.1.8#3	The short circuit protection mechanism shall be designed to avoid tripping at the maximum continuous operating current of the Source port and shall not interfere with negotiated current levels.	Vendor declaration	
7.1.8#4	The protected port shall recover automatically, by performing a Hard Reset, when the fault is removed and resume operation at vSafe5V.	Vendor declaration VEND_GEN	
7.1.8#5	Mechanical or user intervention shall not be required to reset the short circuit protection mechanism.	Vendor declaration VEND_GEN	
7.1.8#6	Safe Operation mandates that Power Delivery Sources shall be tolerant of vSafe5V being present on VBUS when simultaneously applying power to VBUS.	Vendor declaration	
7.1.8#7	This shall not interfere with normal PD communication.		
7.1.8#8	When the Source is capable of detecting a detach either mechanically or electrically, the Source shall transition to vSafe0V by tSafe0V relative to when the detach event occurred (i.e. physical removal of the plug).		
7.1.8#9	During the transition to vSafe0V the VBUS voltage shall be below vSafe5V max by tSafe5V relative to when the detach event occurred (i.e. physical removal of the plug) and shall not exceed vSafe5V max after this time.		
7.1.9 Output Voltage Tolerance and Range			
7.1.9#1	During static load conditions the Source output voltage shall remain within the vSrcNew range.		
7.1.9#2	In response to a transient load event (as described below) the Source output voltage shall not go beyond the range		

R2.0 Assertion #	Description	Test	Comments
	specified by vSrcValid.		
7.1.9#3	The amount of time the Source output voltage can be in the band between vSrcNew and vSrcValid shall not exceed tSrcTransient.		
7.1.9#4	The Source output voltage tolerance, vSrcNew, shall be measured at the connector receptacle.	POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.9#5	The stability of the Source shall be tested in 25% load step increments from minimum load to maximum load and also from maximum load to minimum load	POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.9#6	The time between each step shall be sufficient to allow for the output voltage to settle between load steps.		
7.1.10 Charging and Discharging the Bulk Capacitance on VBUS			
7.1.10#1	The Source shall charge and discharge the bulk capacitance on VBUS whenever the Source voltage is negotiated to a different value.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.10#2	The charging or discharging occurs during the voltage transition and shall not interfere with the Source's ability to meet tSrcReady.	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.11 Swap Standby for Sources			
7.1.11#1	Sources and Sinks of a Dual_Role port shall support Swap Standby.	PROT_PROC_SWAP_TSTR_SNK_2 PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_UUT_SNK_3 PROT_PROC_SWAP_UUT_SRC_3	
7.1.11#2	While in Swap Standby: · The Source shall not drive VBUS, that is therefore expected to remain at vSafe0V. · Any discharge circuitry that was used to achieve vSafe0V shall be removed from VBUS. · The Dual_Role Port shall be configured as a Sink · The USB connection shall not reset even though vSafe5V is no longer present on VBUS (see Section 9.1.2).	PROT_PROC_SWAP_TSTR_SRC_8 PROT_PROC_SWAP_UUT_SNK_4 PROT_PROC_SWAP_TSTR_SNK_3	
7.1.11#3	The PS_RDY message associated with the Source being in Swap Standby shall be sent after the VBUS drive is removed.	PROT_PROC_SWAP_TSTR_SRC_3 PROT_PROC_SWAP_UUT_SNK_5 PROT_PROC_SWAP_TSTR_SNK_5	
7.1.11#4	The time for the Source to transition to Swap Standby shall not exceed tSrcSwapStdby.	PROT_PROC_SWAP_TSTR_SRC_8 PROT_PROC_SWAP_UUT_SNK_4	
7.1.11#5	The transition time from Swap Standby to being the new Sink shall be no more than tNewSnk.	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SNK	
7.1.12 Source Peak Current Operation			
7.1.12#1	A Source that has the Fixed Supply PDO Peak Current bits set to 01b, 10b and 11b shall be designed to support one of the overload capabilities defined in Table 6 7.	not tested	could invent test?
7.1.13.1 Noise Injected on VBUS			
7.1.13.1#1	The Source shall not interfere with the USB Power Delivery BFSK waveform that is transmitted on VBUS.	POW_SNK_NOIS_C_CP	
7.1.13.1#2	Out_of_band noise (including spurs) can also affect the reception of the BFSK signal; therefore the Source shall limit its out_of_band noise below the spectrum shown in Figure 7 5 and the levels indicated in Table 7 1.	POW_SNK_NOIS_C_CP	
7.1.13.1#3	The VBUS_to_GND noise measurement shall be made at the connector.	POW_SNK_NOIS_C_CP	
7.1.13.3 Dead Battery Operation			
7.1.13.3#1	Dead Battery operation shall be supported by all Consumer/Providers.		
7.1.13.3#2	A Consumer/Provider shall apply vSafeDB to its port when it detects its port voltage to be vSafe0V (see Section 4.1).		

R2.0 Assertion #	Description	Test	Comments
7.1.13.3#3	The Operation Region of vSafeDB source shall be as shown in Figure 7 6.		
7.1.13.3#4	The Consumer/Provider shall fully support the vSafeDB Operating Region as shown in Figure 7 6 on reaching 2V.		
7.1.13.3#5	The output voltage shall reach 2V and comply with the vSafeDB Operating Region within tTurnOnSafeDB of when it first applies voltage to VBUS.		
7.1.13.3#6	The Source bulk capacitance shall be limited to cSrcBulkDB for the Consumer/Provider when applying vSafeDB.		
7.1.13.3#7	The Consumer/Provider shall be limited to cSnkBulk when not applying vSafeDB.		
7.2.1 Behavioral Aspects			
7.2.1#1	The Sink in the USB Power Delivery system shall exhibit the following behaviors. · Shall be backward compatible with legacy VBUS ports.	USB2 and USB3 testing	
7.2.1#2	"The Sink in the USB Power Delivery system shall exhibit the following behaviors. _Shall draw the default [USB2.0], [USB3.1], [USBType_C1.0] or [BC1.2] VBUS current when the USB cable is attached (USB Default Operation)."	USB2 and USB3 testing	
7.2.1#3	"The Sink in the USB Power Delivery system shall exhibit the following behaviors. Shall draw the default [USB2.0], [USB3.1], [USBType_C1.0] or [BC1.2] VBUS current when a USB Power Delivery Contract does not exist (USB Default Operation)."	USB2 and USB3 testing	
7.2.1#4	"The Sink in the USB Power Delivery system shall exhibit the following behaviors. Shall return to the default [USB2.0], [USB3.1], [USBType_C1.0] or [BC1.2] VBUS when responding to a Hard Reset (USB Default Operation)."		
7.2.1#5	The Sink in the USB Power Delivery system shall exhibit the following behaviors. · Shall control VBUS in_rush current when increasing current consumption.		
7.2.1#6	The Sink in a Provider/Consumer exhibits the following behaviors: • Shall return to Source operation when responding to a Hard Reset		
7.2.1#7	The Sink in a Provider/Consumer exhibits the following behaviors: • Shall control VBUS in_rush current when increasing and decreasing current consumption		
7.2.2 Sink Bulk Capacitance			
7.2.2#1	The Sink shall have a bulk capacitance, cSnkBulk, located between the input of the power supply and the transceiver isolation impedance as shown in Figure 7_7.	Vendor Declaration	
7.2.2#2	The Sink bulk capacitance shall not be placed between the transceiver isolation impedance and the USB receptacle.	PHYFSK_TX_LEV PHYFSK_RX_BER	
7.2.2#3	An upper bound of cSnkBulkPd shall not be exceeded	Vendor Declaration	
7.2.2#4	Regardless of when the change occurs, the capacitance change shall occur in such a manner that does not introduce a VBUS transient current greater than iCapChange.	Vendor Declaration	
7.2.2#5	During a Power Role Swap the Default Sink shall transition to Swap Standby before operating as the new Source.	PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_UUT_SNK_3	
7.2.2#6	Any change in bulk capacitance required to complete the Power Role Swap shall occur during Swap Standby.	Vendor Declaration	
7.2.3 Sink Standby			

R2.0 Assertion #	Description	Test	Comments
7.2.3#1	The Sink shall transition to Sink Standby before a positive or negative voltage transition of VBUS.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	maybe also in PROT_PROC_
7.2.3#2	During Sink Standby the Sink shall reduce its average power draw to pSnkStdby and its peak power draw shall not exceed pSnkStdbyLimit.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	maybe also in PROT_PROC_
7.2.3#3	The Sink shall complete this transition to Sink Standby within tSnkStdby after evaluating the Accept Message from the Source.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	maybe also in PROT_PROC_
7.2.3#4	The transition when returning to Sink operation from Sink Standby shall be completed within tSnkRtn.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	maybe also in PROT_PROC_
7.2.3#5	The pSnkStdby requirement shall only apply if the Sink power draw is higher than this level.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	maybe also in PROT_PROC_
7.2.4 Suspend Power Consumption			
7.2.4#1	When Source has set its USB Suspend Supported flag (see Section 6.4.1.2.2.2), a Sink shall go to the lowest power state during USB suspend.		
7.2.4#2	The lowest power state shall be pSnkSusp or lower for a PDUSB peripheral device and pHubSusp or lower for a PDUSB Hub.	PROT_PROC_SWAP_TSTR_SNK_6	
7.2.5 Zero Negotiated Current			
7.2.5#1	When a Sink Requests zero current as part of a power negotiation with a Source the Sink shall go to the lowest power state, pSnkSusp or lower, where it can still communicate using PD signaling.		
7.2.6 Transient Load Behavior			
7.2.6#1	When a Sink’s operating current changes due to a load step, load release or any other change in load level, the positive or negative overshoot of the new load current shall not exceed the range defined by iOvershoot.		
7.2.6#2	The rate of change of any shift in Sink load current during normal operation shall not exceed iLoadStepRate (for load steps) and iLoadReleaseRate (for load releases) as measured at the Sink receptacle.		
7.2.7 Swap Standby for Sinks			
7.2.7#1	The Sink capability in a Dual_Role port shall support Swap Standby.	PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_UUT_SNK_3	
7.2.7#2	While in Swap Standby the Sink's current draw shall not exceed iSnkSwapStdby from VBUS	PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_UUT_SNK_3	
7.2.7#3	While in Swap Standby...the Dual_Role port shall be configured as a Source after VBUS has been discharged to vSafe0V by the existing Initial Source.	PROT_PROC_SWAP_TSTR_SRC_8 PROT_PROC_SWAP_UUT_SNK_4	
7.2.7#4	The time for the Sink to transition to SwapStandby shall be no more than tSnkSwapStdby.	PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_UUT_SNK_4 PROT_PROC_SWAP_TSTR_SNK_2	
7.2.7#5	The transition time from Swap Standby to new Source shall be no more than tNewSRC.	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SNK	
7.2.8 Sink Peak Current Operation			
7.2.8#1	Sinks shall only make use of a Source overload capability when the corresponding Fixed Supply PDO Peak Current bits are set to 01b, 10b and 11b.	not tested	
7.2.8#2	Sinks shall manage thermal aspects of the overload event by not exceeding the average negotiated output of a Fixed Supply that supports Peak Current operation.	not tested	
7.2.8#3	Sinks that depend on the Peak Current capability for enhanced system performance shall also function correctly when attached to a Source that does not offer the Peak Current capability or when the Peak Current capability has been inhibited by the Source.	not tested	

R2.0 Assertion #	Description	Test	Comments
7.2.9.2 Noise Reflected on VBUS			
7.2.9.2#1	The Sink shall not interfere with the USB Power Delivery BFSK waveform that is transmitted on VBUS.	POW_SNK_NOIS_C_CP	
7.2.9.2#2	Out_of_band noise (including spurs) can also affect the reception of the BFSK signal; therefore the Sink shall limit its out_of_band noise below the spectrum shown in Figure 7 8 and as detailed in Table 7 2.	POW_SNK_NOIS_C_CP	
7.2.9.2#3	Figure 7 8 also shows the noise level that shall be allowed relative to the level of the allowed in_band noise [for a Sink: dbV(minimum value of vTX) – snrSnk].	POW_SNK_NOIS_C_CP	
7.2.9.2#4	The VBUS_to_GND noise measurement shall be made at the connector.	POW_SNK_NOIS_C_CP	
7.2.9.3 Dead Battery Operation			
7.2.9.3#1	When a Provider/Consumer port is not able to Source power (or has chosen to not source power) and would like to be powered by a Consumer/Provider, the Provider/Consumer shall discharge its port voltage to vSafe0V, present cSnkBulk and be ready to power up the transceiver if vSafeDB is applied to its port.		
7.2.9.3#2	When operating from vSafeDB, the Provider/Consumer, acting as Sink, shall be able to operate from any Source that complies with the Operating Region of Figure 7 6		
7.2.9.3#3	The Provider/Consumer may regain the ability to Source power (or decide to do so) at any time and shall not apply power to its port when vSafeDB is present.		
7.2.9.3#4	The Provider/Consumer shall not commence Dead Battery operation or draw more than 1 mA until the voltage on VBUS is above 2V.		
7.2.9.3#5	The Sink bulk capacitance shall be limited to cSnkBulkDB for the Provider/Consumer acting as the Sink during Dead Battery operation.		
7.2.10 Safe Operating Considerations			
7.2.10#1	When a Source is detached from a Sink, the Sink shall continue to draw power from its input bulk capacitance until VBUS is discharged to vSafe5V or lower by no longer than tSafe5V from the detach event.		
7.2.10#2	This safe Sink requirement shall apply to all Sinks operating with a negotiated VBUS level greater than vSafe5V and shall apply during all low power and high power operating modes of the Sink.		
7.3 Transitions			
7.3#1	The following types of Power Transitions shall also be applied when moving from [USB2.0], [USB3.1], [USBType_C1.0] or [BC1.2] operation into Power Delivery Mode: • High Power • Relatively Constant Power • Lower Power Transitions • No change in Current or Voltage.		
7.3.1 Increasing the Current			
7.3.1#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed when increasing the current is shown in Figure 7 9.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.1#2	The sequence that shall be followed is described in Table 7 3.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.1#3	The timing parameters that shall be followed are listed in Table 7 22 and Table 7 23.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.1#4	The Power Supply shall be ready to operate at the new power level within tSrcReady (t1).	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	

R2.0 Assertion #	Description	Test	Comments
7.3.1#5	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.2 Increasing the Voltage			
7.3.2#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed when increasing the voltage is shown in Figure 7 10.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.2#2	The sequence that shall be followed is described in Table 7 5.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.2#3	The timing parameters that shall be followed are listed in Table 7 22, Table 7 23, and Table 7 24.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.2#4	Policy Engine tells the Device Policy Manager to instruct the Power Supply to reduce power consumption to pSnkStdby within tSnkStdby (t1); t1 shall complete before tSrcTransition.		
7.3.2#5	The Sink shall not violate transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.2#6	The Power Supply shall be ready to operate at the new power level within tSrcReady (t2).	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.2#7	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.3 Increasing the Voltage and Current			
7.3.3#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed when increasing the voltage and current is shown in Figure 7 14.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.3#2	The sequence that shall be followed is described in Table 7 5.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.3#3	The timing parameters that shall be followed are listed in Table 7 22, Table 7 23, and Table 7 24.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.3#4	"Policy Engine tells the Device Policy Manager to instruct the power supply to reduce power consumption to pSnkStdby within tSnkStdby (t1) ; t1 shall complete before tSnkTransition."		
7.3.3#5	The Sink shall not violate transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.3#6	The Power Supply shall be ready to operate at the new power level within tSrcReady (t2).	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.3#7	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.4 Increasing the Voltage and Decreasing the Current			
7.3.4#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed when increasing the voltage and decreasing the current is shown in Figure 7 12.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.4#2	The sequence that shall be followed is described in Table 7 6.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.4#3	The timing parameters that shall be followed are listed in Table 7 22, Table 7 23, and Table 7 24.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.4#4	Policy Engine tells the Device Policy Manager to instruct the power supply to reduce power consumption to pSnkStdby within tSnkStdby (t1) ; t1 shall complete before tSnkTransition.		
7.3.4#5	The Sink shall not violate transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.4#6	The Power Supply shall be ready to operate at the new power	POW_SNK_TRANS_C_CP	

R2.0 Assertion #	Description	Test	Comments
	level within tSrcReady (t2).	POW_SNK_TRANS_PC	
7.3.4#7	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.5 Decreasing the Voltage and Increasing the Current			
7.3.5#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed when decreasing the voltage and increasing the current is shown in Figure 7 13.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.5#2	The sequence that shall be followed is described in Table 7 8.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.5#3	The timing parameters that shall be followed are listed in Table 7 22, Table 7 23, and Table 7 24.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.5#4	Policy Engine tells the Device Policy Manager to instruct the Power Supply to reduce power consumption to pSnkStdby within tSnkStdby (t1) ; t1 shall complete before tSrcTransition.		
7.3.5#5	The Sink shall not violate transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.5#6	The Power Supply shall be ready to operate at the new power level within tSrcReady (t2).	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.5#7	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.6 Decreasing the Current			
7.3.6#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed when decreasing the current is shown in Figure 7 17.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.6#2	The sequence that shall be followed is described in Table 7 8.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.6#3	The timing parameters that shall be followed are listed in Table 7 22, Table 7 23, and Table 7 24.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.6#4	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.6#5	The Sink shall be able to operate with lower currentwithin tSnkNewPower (t1)		
7.3.6#6	The Power Supply shall be ready to operate at the new power level within tSrcReady (t2).	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.7 Decreasing the Voltage			
7.3.7#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed when decreasing the voltage is shown in Figure 7 15.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.7#2	The sequence that shall be followed is described in Table 7 10.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.7#3	The timing parameters that shall be followed are listed in Table 7 22, Table 7 23, and Table 7 24.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.7#4	Policy Engine tells the Device Policy Manager to instruct the Power Supply to reduce power consumption to pSnkStdby within tSnkStdby (t1) ; t1 shall complete before tSnkTransition		
7.3.7#5	The Sink shall not violate transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.7#6	The Power Supply shall be ready to operate at the new power level within tSrcReady (t2).	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.7#7	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	

R2.0 Assertion #	Description	Test	Comments
	operating at the new power level.		
7.3.8 Decreasing the Voltage and the Current			
7.3.8#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed when decreasing the voltage and current is shown in Figure 7 16.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.8#2	The sequence that shall be followed is described in Table 7 11.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.8#3	The timing parameters that shall be followed are listed in Table 7 22, Table 7 23, and Table 7 24.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.8#4	The Sink shall not violate transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.8#5	The Power Supply shall be ready to operate at the new power level within tSrcReady (t2).	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.8#6	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.9 Sink Requested Role Swap			
7.3.9#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed during a Sink requested Role Swap is shown in Figure 7 17.	PROT_PROC_SWAP_TSTR_SNK PROT_PROC_SWAP_UUT_SNK	
7.3.9#2	The sequence that shall be followed is described in Table 7 12.	PROT_PROC_SWAP_TSTR_SNK PROT_PROC_SWAP_UUT_SNK	
7.3.9#3	The timing parameters that shall be followed are listed in Table 7 24.	PROT_PROC_SWAP_TSTR_SNK PROT_PROC_SWAP_UUT_SNK	
7.3.9#4	When in Sink Standby the Initial Sink shall not draw more than iSnkSwapStdby (I1).	PROT_PROC_SWAP_TSTR_SNK_2 PROT_PROC_SWAP_UUT_SNK_3	
7.3.9#5	The Sink shall not violate transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.9#6	The Power Supply shall complete the transition to Swap Standby within tSrcSwapStdby (t2).	PROT_PROC_SWAP_TSTR_SNK_3 PROT_PROC_SWAP_UUT_SNK_4	
7.3.9#7	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.10 Source Requested Role Swap			
7.3.10#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed during a Source requested Power Role Swap is shown in Figure 7 18.	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC	
7.3.10#2	The sequence that shall be followed is described in Table 7 13.	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC	
7.3.10#3	The timing parameters that shall be followed are listed in Table 7 22.	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC	
7.3.10#4	The Power supply shall complete the transition to Swap Standby within tSnkStdby (t1).	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC_4	
7.3.10#5	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.10#6	When in Sink Standby the Initial Sink shall not draw more than iSnkSwapStdby (I1).	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC_3	
7.3.10#7	The Power Supply shall complete the transition to Swap Standby within tSrcSwapStdby (t2).	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC_4	
7.3.10#8	The new Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating atthe new power level.		
7.3.11 GotoMin Current Decrease			
7.3.11#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed during a GotoMin	PROT_PROC_GOTOMIN_TSTR	

R2.0 Assertion #	Description	Test	Comments
	current decrease is shown in Figure 7 19.		
7.3.11#2	The sequence that shall be followed is described in Table 7 19.	PROT_PROC_GOTOMIN_TSTR	
7.3.11#3	The timing parameters that shall be followed are listed in Table 7 22 and Table 7 13.	PROT_PROC_GOTOMIN_TSTR	
7.3.11#4	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.11#5	The Power Supply shall be ready to operate at the new power level within tSrcReady (t2).	PROT_PROC_GOTOMIN_TSTR	
7.3.12 Source Initiated Hard Reset			
7.3.12#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed during a Source Initiated Hard Reset is shown in Figure 7 20.		
7.3.12#2	The sequence that shall be followed is described in Table 7 15.		
7.3.12#3	The timing parameters that shall be followed are listed in Table 7 22 and Table 7 23.		
7.3.12#4	The Sink prepares for the Hard Reset within tSnkHardResetPrepare (t1)) and passes an indication to the Device Policy Manger The Sink shall not draw more than iSafe0mA when VBUS is driven to vSafe0V.		
7.3.12#5	The transition to vSafe0V shall occur within tSafe0V (t1).	[PROT_PROC_HR_TSTR_2]	
7.3.12#6	The transition to vSafe5V shall occur within tSrcTurnOn.	[PROT_PROC_HR_TSTR_5]	
7.3.12#7	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.13 Sink Initiated Hard Reset			
7.3.13#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed during a Sink Initiated Hard Reset is shown in Figure 7 21.		
7.3.13#2	The sequence that shall be followed is described in Table 7 16.		
7.3.13#3	The timing parameters that shall be followed are listed in Table 7 22 and Table 7 23.		
7.3.13#4	The Sink prepares for the Hard Reset within tSnkHardResetPrepare (t1) and passes an indication to the Device Policy Manger The Sink shall not draw more than iSafe0mA when VBUS is driven to vSafe0V.		
7.3.13#5	The transition shall occur within tSafe0V (t2).		
7.3.13#6	The Sink shall not draw more than iSafe0mA when VBUS is driven to vSafe0V.		
7.3.13#7	The transition to vSafe5V shall occur within tSrcTurnOn (t3).		
7.3.13#8	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.14 New Source Initiates Hard Reset and New Sink Receives Hard Reset Signaling			
7.3.14#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed during a Hard Reset initiated by the Type-B new Source when the new Type-A Sink receives Hard Reset signaling is shown in Figure 7 25.		
7.3.14#2	The sequence that shall be followed is described in Table 7 16.		
7.3.14#3	The timing parameters that shall be followed are listed in Table 7 22 and Table 7 23.		
7.3.14#4	The transition to vSafe5V shall be completed with tSrcTurnOn		

R2.0 Assertion #	Description	Test	Comments
	(t2).		
7.3.14#5	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.15 New Source Initiates Hard Reset and New Sink Does Not Receive Hard Reset Signaling			
7.3.15#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed during a Hard Reset initiated by the Type-B new Source when the new Type-A Sink does not receive the Hard Reset signaling is shown in Figure 7 26.		
7.3.15#2	The sequence that shall be followed is described in Table 7 17.		
7.3.15#3	The timing parameters that shall be followed are listed in Table 7 22 and Table 7 23.		
7.3.15#4	Source shall not re_apply vSafe5V until VBUS is within vSafe0V.		
7.3.15#5	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.15#6	The transition to vSafe5V shall be completed with tSrcTurnOn(t2).		
7.3.16 New Sink Initiates Hard Reset and New Source Receives Hard Reset Signaling			
7.3.16#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed during a Hard Reset initiated by the Type-A new Sink when the Type-B new Source receives the Hard Reset signaling is shown in Figure 7 27.		
7.3.16#2	The sequence that shall be followed is described in Table 7 18.		
7.3.16#3	The timing parameters that shall be followed are listed in Table 7 22 and Table 7 23.		
7.3.16#4	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.16#5	The transition to vSafe5V shall be completed with tSrcTurnOn (t2).		
7.3.17 New Sink Initiates Hard Reset and New Source Does Not Receive Hard Reset Signaling			
7.3.17#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed during a Hard Reset initiated by the Type-A new Sink when the Type-B new Source does not receive the Hard Reset signaling is shown in Figure 7 28.		
7.3.17#2	The sequence that shall be followed is described in Table 7 19.		
7.3.17#3	The timing parameters that shall be followed are listed in Table 7 22 and Table 7 23.		
7.3.17#4	Source shall not re_apply vSafe5V until VBUS is within vSafe0V.		
7.3.17#5	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.17#6	The transition to vSafe5V shall be completed with tSrcTurnOn (t2).		
7.3.18 Type_A to Type_B Dead Battery Operation			
7.3.18#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed during Type_A to Type_B Dead Battery operation is shown in Figure 7 26.	POW_DB_PC POW_DB_CP	
7.3.18#2	The sequence that shall be followed is described in Table 7	POW_DB_PC	

R2.0 Assertion #	Description	Test	Comments
	20.	POW_DB_CP	
7.3.18#3	The timing parameters that shall be followed are listed in Table 4 4, Table 7 22 and Table 7 23.	POW_DB_PC POW_DB_CP	
7.3.18#6	The Sink shall not violate the transient load behavior defined in Section 7.2.6 while transitioning to and operating at the new power level.		
7.3.19 No Change in Current or Voltage			
7.3.19#1	The interaction of the System Policy, Device Policy, and Power Supply that shall be followed when the Sink requests the same Voltage and Current as it is currently operating at is shown in Figure 7 27.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.19#2	The sequence that shall be followed is described in Table 7 21	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.19#3	The timing parameters that shall be followed are listed in Table 7 22 and Table 7 23.	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.4.1 Source Electrical Parameters			
7.4.1#1	The Source Electrical Parameters that shall be followed are specified in Table 7 22 and Table 4 4.		
7.4.1#2	Note 1: The Source shall charge and discharge the total bulk capacitance to meet the transition time requirments.		
7.4.2 Sink Electrical Parameters			
7.4.1#1	The Sink Electrical Parameters that shall be followed are specified in Table 7 23 and Table 4 4.		
7.4.3 Common Electrical Parameters			
7.4.1#1	The Sink Electrical Parameters that shall be followed are specified in Table 7 24		

7.6.2 Recommended (Should)

R2.0 Assertion #	Description	Test	Comments
Subsection 7.1 Source Requirements			
Subsection 7.1.6 Response to Hard Resets			
7.1.6#r1	• Self_powered devices should not disconnect from USB during a Hard Reset.		
7.1.11 Output Voltage Tolerance			
7.1.11#r1	The time between each step should be sufficient to allow for the output voltage to settle in between load steps.		
7.1.13.1 Transceiver Isolation Impedance			
7.1.13.1#r1	The Source output impedance, the total VBUS bulk capacitance, the transceiver isolation impedance(s), the USB cable and the Sink network creates a complex output isolation impedance that should be taken into consideration when designing the Source.		
7.2.7 Swpa Standby for Sinks			
7.2.7#r1	The Sink's USB connection should not be reset even though vSafe5V is not present on the VBUS conductor (see Section 9.1.2).		
7.2.9.1 Transceiver Isolation Impedance			
7.2.9.1#r1	The Sink input impedance, the total VBUS bulk capacitance, the transceiver isolation impedance(s), the USB cable and the Source creates a complex input isolation impedance that should be taken into consideration when designing the Sink.		

7.7 Chapter 8 Device Policy

7.7.1 Normative (Shall/Normative)

R2.0 Assertion #	Description	Test	Comments
Subsection 8.2 Device Policy Manager			
Subsection 8.2.1 Capabilities			
8.2.1#1	The Device Policy Manager in a Provider shall know the Power supplies available in the Device and their capabilities.		
8.2.1#2	In addition it shall be aware of any other PD Sources of power such as batteries and AC inputs.		
8.2.1#3	The available power sources and existing demands on the Device shall be taken into account when presenting capabilities to a Sink.		
8.2.1#4	The Device Policy Manager in a Consumer shall know the requirements of the Sink and use this to evaluate the capabilities offered by a Source.		
8.2.1#5	It shall be aware of its own power sources e.g. Batteries or AC supplies where these have a bearing on its operation as a Sink.		
8.2.1#6	The Device Policy Manager in a Provider/Consumer or Consumer/Provider shall combine the above capabilities		
8.2.1#7	The Device Policy Manager in a Provider/Consumer or Consumer/Provider...shall also be able to present the dual_role nature of the Device to an attached PD Device.		
8.2.2 System Policy			
8.2.2#1	A given PD device may have no USB capability, or PD may have been added to a USB device in such a way that PD is not integrated with USB. In these two cases there shall be no requirement for the Device Policy Manager to interact with the USB interface of the device.		
8.2.2#2	The following requirements shall only apply to PD devices that expose PD functionality over USB.		
8.2.2#3	The Device Policy Manager shall communicate over USB with the System Policy Manager according to the requirements detailed in Chapter 9.		
8.2.2#4	Whenever requested the Device Policy Manager shall implement a Local Policy according to that requested by the System Policy Manager.		
8.2.2#5	Note: that due to timing constraints, a PD Capable device shall be able to respond autonomously to all time_critical PD related requests.		
8.2.3 Control of Source/Sink			
8.2.3#1	The Device Policy Manager for a Provider shall manage the power supply for each PD Source Port		
8.2.3#2	The Device Policy Manager for a Provider...shall know at any given time what the negotiated power is.		
8.2.3#3	It shall request transitions of the supply and inform the Policy Engine whenever a transition completes.		
8.2.3#4	The Device Policy Manager for a Consumer shall manage the Sink for each PD Sink Port		
8.2.3#5	The Device Policy Manager for a Consumer...shall know at any given time what the negotiated power is.		
8.2.3#6	The Device Policy Manager for a Provider/Consumer or Consumer/Provider shall manage the transition between Source/Sink roles for each PD dual_role port		
8.2.3#7	The Device Policy Manager for a Provider/Consumer or Consumer/Provider...shall know at any given time what operational role the port is in.		
8.2.3#8	The Device Policy Manager for a Consumer/Provider shall be able to detect and handle cases where back_powering is		

R2.0 Assertion #	Description	Test	Comments
	required due to a dead battery on the Provider/Consumer side.		
8.2.3#9	It shall determine the absence of VBUS and the Provider/Consumer's ability to be back_powered from the Cable Detection module.		
8.2.3#10	It shall then initiate Provider role and instruct the Power Supply to start providing default output power.		
8.2.3#11	Where supported it shall determine the presence of VBUS from the Cable Detection module.		
8.2.3#12	It shall then initiate Consumer role and instruct the Power Supply to start sinking default input power.		
8.2.4.1 Device Policy Manager in a Provider			
8.2.4.1#1	The Device Policy Manager in the Provider shall control the Cable Detection module		
8.2.4.1#2	The Device Policy Manager in the Provider... shall be able to use the Cable Detection module to determine the attachment of a cable and for Type_A/Type_B whether it is a USB Power Delivery or non_USB Power Delivery cable.		
8.2.4.1#3	This information and the type of receptacle on the local device shall be used to determine the capabilities of the port and attached cabling.		
8.2.4.2 Device Policy Manager in a Consumer			
8.2.4.2#1	The Device Policy Manager in a Consumer...shall be able to use the Cable Detection module to determine the attachment of a USB Power Delivery or non_USB Power Delivery cable.		
8.2.4.3 Device Policy Manager in a Consumer/Provider			
8.2.4.3#1	The Device Policy Manager in a Consumer/Provider...shall control the Cable Detection module in order to support the Dead Battery back_powering case to determine the following for a given port: · Attachment of a USB Power Delivery Provider/Consumer which supports Dead Battery back_powering		
8.2.4.3#2	The Device Policy Manager in a Consumer/Provider...shall control the Cable Detection module in order to support the Dead Battery back_powering case to determine the following for a given port: · Presence of VBUS.		
8.2.5 Managing Power Requirements			
8.2.5#1	The Device Policy Manager in a Provider shall be aware of the power requirements of all Devices connected to its Source Ports.		
8.2.5#2	The Consumer's Device Policy Manager shall ensure that it takes no more power than is required to perform its functions and gives back unneeded power whenever possible		
8.2.5#3	(in such cases the Provider shall maintain a Power reserve to ensure future operation is possible).		
8.2.5.1 Managing the Power Reserve			
8.2.5.1#1	It shall be the Device Policy Manager's responsibility to allocate power and maintain a power Reserve so as not to over_subscribe its available power resource.		
8.2.5.1#2	A Device with multiple ports such as a Hub shall always be able to meet the incremental demands of the port requiring the highest incremental power from its Power Reserve.		
8.2.5.1#3	A Consumer requesting power shall take into account its operational requirements when advertising its ability to temporarily return power.		

R2.0 Assertion #	Description	Test	Comments
8.2.5.1#4	When the Consumer's requirements change, it shall re_negotiate its power to reflect the changed requirements.		
8.2.5.2.1 Local device handling of mismatch			
8.2.5.2.1#1	The Consumer's Device Policy Manager shall cause a message to be displayed to the end user that a power capability mismatch has occurred.		
8.2.5.2.2 Device Policy Manager Communication with System Policy			
8.2.5.2.2#1	In a USB Power Delivery aware system with an active System Policy manager (see Section 8.2.2), the Device Policy Manager shall notify the System Policy Manager of the mismatch.		
8.2.5.2.2#2	This information shall be passed back to the System Policy Manager using the mechanisms described in Chapter 8.3.3.		
8.2.5.2.2#3	In order to identify a more suitable Source Port for the Consumer the System Policy Manager shall communicate with the Device Policy Manager in order to determine the Consumer's requirements.		
8.2.5.2.2#4	The Device Policy Manager shall use a Get_Sink_Cap message (see Section 6.3.8) to discover which Power levels can be utilized by the Consumer.		
8.2.6 Use of "Externally Powered" bit with Batteries and AC supplies			
8.2.6#1	When monitored, and a USB interface is supported, the External Power status (see Section 9.4.4) and the battery state (see Section 9.4.2) shall also be reported to the System Policy Manager using the USB interface.		
8.2.6.2 Battery supplies			
8.2.6.2#1	When monitored, and a USB interface is supported, the battery state shall be reported to the System Policy Manager using the USB interface.		
8.2.6.2#2	1. Provider/Consumers using external sources ("Externally powered" bit set) shall always deny Power Role swap requests from Consumer/Providers not using external sources ("Externally Powered" bit cleared).	PROT_SEQ_SWAP_REJ	
8.2.6.2#3	2. Provider/Consumers not using external sources ("Externally Powered" bit cleared) shall always accept a Power Role swap request from a Consumer/Provider using external power sources ("Externally Powered" bit set) unless the requester is not able to provide the requirements of the present Provider/Consumer.		
8.2.7 Interface to the Policy Engine			
8.2.7#1	The Device Policy Manager shall maintain an interface to the Policy Engine for each port in the device.		
8.2.7.1 Device Policy Manager in a Provider			
8.2.7.1#1	The Device Policy Manager in a Provider shall also provide the following functions to the Policy Engine: · Inform the Policy Engine of changes in cable/Device attachment status for a given cable.		
8.2.7.1#2	The Device Policy Manager in a Provider shall also provide the following functions to the Policy Engine: · Inform the Policy Engine whenever the Source capabilities available for a port change.		
8.2.7.1#3	The Device Policy Manager in a Provider shall also provide the following functions to the Policy Engine: · Evaluate requests from an attached Consumer and provide responses to the Policy Engine.		
8.2.7.1#4	The Device Policy Manager in a Provider shall also provide the following functions to the Policy Engine: · Respond to requests for Power Supply transitions from the Policy Engine.		

R2.0 Assertion #	Description	Test	Comments
8.2.7.1#5	The Device Policy Manager in a Provider shall also provide the following functions to the Policy Engine: · Indication to Policy Engine when Power Supply transitions are complete.		
8.2.7.1#6	The Device Policy Manager in a Provider shall also provide the following functions to the Policy Engine: · Maintain reserve capability for Devices operating on a port at less than maximum power.		
8.2.7.2 Device Policy Manager in a Consumer			
8.2.7.2#1	The Device Policy Manager in a Consumer shall also provide the following functions to the Policy Engine: · Inform the Policy Engine of changes in cable/Device attachment status.		
8.2.7.2#2	The Device Policy Manager in a Consumer shall also provide the following functions to the Policy Engine: · Inform the Policy Engine whenever the power requirements for a port change.		
8.2.7.2#3	The Device Policy Manager in a Consumer shall also provide the following functions to the Policy Engine: • Evaluate Source capabilities and provide suitable responses: o Request from offered capabilities. o Indicate whether additional power is required.		
8.2.7.2#4	The Device Policy Manager in a Consumer shall also provide the following functions to the Policy Engine: · Respond to requests for Sink transitions from the Policy Engine.		
8.2.7.3 Device Policy Manager in a Provider/Consumer			
8.2.7.3#1	The Device Policy Manager in a Provider/Consumer shall provide the following functions to the Policy Engine: · Provider Device Policy Manager		
8.2.7.3#2	The Device Policy Manager in a Provider/Consumer shall provide the following functions to the Policy Engine: · Consumer Device Policy Manager		
8.2.7.3#3	The Device Policy Manager in a Provider/Consumer shall provide the following functions to the Policy Engine: · Interface for the Policy Engine to request Power Supply transitions from Source to Sink and vice versa.		
8.2.7.3#4	The Device Policy Manager in a Provider/Consumer shall provide the following functions to the Policy Engine: · Indications to Policy Engine during Power role Swap transitions.		
8.2.7.4 Device Policy Manager in a Provider/Consumer dead battery handling			
8.2.7.4#1	• In this scenario a Provider/Consumer shall: • Detect that VBUS is present and that its battery is dead		
8.2.7.4#2	• In this scenario a Provider/Consumer shall: • Switch to Consumer role without using a PR_Swap message		
8.2.7.4#3	• In this scenario a Provider/Consumer shall: • Use VBUS to power the USB Power Delivery communications		
8.2.7.5 Device Policy Manager in a Consumer/Provider			
8.2.7.5#1	The Device Policy Manager in a Consumer/Provider shall provide the following functions to the Policy Engine: · Consumer Device Policy Manager		
8.2.7.5#2	The Device Policy Manager in a Consumer/Provider shall provide the following functions to the Policy Engine: · Provider Device Policy Manager		
8.2.7.5#3	The Device Policy Manager in a Consumer/Provider shall provide the following functions to the Policy Engine:		

R2.0 Assertion #	Description	Test	Comments
	· Interface for the Policy Engine to request Power Supply transitions from Sink to Source and vice versa.		
8.2.7.5#4	The Device Policy Manager in a Consumer/Provider shall provide the following functions to the Policy Engine: · Indications to Policy Engine during Power role Swap transitions.		
8.2.7.6 Device Policy Manager in a Consumer/Provider dead battery handling			
8.2.7.6#1	The Device Policy Manager in a Consumer/Provider shall also provide: • Detection and handling of back powering in the case of a dead battery (see Section 4.1).		
8.2.7.6#2	In this scenario a Consumer/Provider shall: · Detect that a Provider/Consumer with a Dead Battery capable of being back powered is attached · Check that VBUS is not present. · Temporarily output vSafeDB on VBUS to provide power to the Provider/Consumer so that it can send a Bit Stream. · Detect Bit Stream 1 sent by Provider/Consumer. · Switch to the Provider role without using a PR_Swap message and output vSafe5V on VBUS that will power the Provider/Consumer's PD communications.		
8.3.2.6 Hard Reset			
8.3.2.6#1	During the Hard Reset USB Power Delivery PHY Layer communications shall be disabled preventing communication between the Port partners.		
8.3.3.1 Introduction to state diagrams used in Chapter 8			
8.3.3.2#1	The state diagrams defined in Section 8.3.3 are normative and shall define the operation of the Power Delivery Policy Engine.		
8.3.3.2 Policy Engine Source Port State Diagram			
8.3.3.2#1	In the case where this is not implemented the Source shall continue to send Capabilities messages each time the SourceCapabilityTimer times out.		
8.3.3.2.1 PE_SRC_Startup			
8.3.3.2.1#1	PE_SRC_Startup shall be the starting state for a Source Policy Engine either on power up or after a Hard Reset.		
8.3.3.2.1#2	On entry to this state the Policy Engine shall reset the CapsCounter and reset the Protocol Layer.		
8.3.3.2.1#3	The Policy Engine shall transition to the PE_SRC_Send_Capabilities state: · When theProtocol Layer reset has completed if the PE_SRC_Startup state was entered due to the system first starting up.		
8.3.3.2.1#4	The Policy Engine shall transition to the PE_SRC_Send_Capabilities state: · When the SwapSourceStartTimer times out if the PE_SRC_Startup state was entered as the result of a Power Role Swap.		
8.3.3.2.1#5	Note: Providers or Provider/Consumers with an insertion detection mechanism (Type_C, Standard_A insertion detect or Micro_A ID pin or Attach Detection Protocol see [USBOTG2.0]) and without a plug attached shall remain in the PE_SRC_Startup state, without sending any Source Capabilities messages until a plug is attached.		
8.3.3.2.2 PE_SRC_Discovery			
8.3.3.2.2#1	On entry to the PE_SRC_Discovery state the Policy Engine shall initialize and run the SourceCapabilityTimer in order to trigger sending a Source Capabilities message.		

R2.0 Assertion #	Description	Test	Comments
8.3.3.2.2#2	The Policy Engine shall transition to the PE_SRC_Send Capabilities state when: · The SourceCapabilityTimer times out and CapsCounter ≤ nCapsCount.		
8.3.3.2.2#3	The Policy Engine shall go to the ErrorRecovery state when: · This is a Type_C Port. · And the NoResponseTimer times out. · And the HardResetCounter > nHardResetCount.		
8.3.3.2.3 PE_SRC_Send Capabilities			
8.3.3.2.3#1	On entry to the PE_SRC_Send Capabilities state the Policy Engine shall request the present port capabilities from the Device Policy Manager.		
8.3.3.2.3#2	The Policy Engine shall then request the Protocol Layer to send a Source Capabilities message containing these capabilities and increment the CapsCounter (if implemented).		
8.3.3.2.3#3	If a GoodCRC message is received then the Policy Engine shall _Stop the NoResponseTimer _Reset the HardResetCounter and CapsCounter to zero. _Initialize and run the SenderResponseTimer.		
8.3.3.2.3#4	Note that the HardResetCounter shall only be set to zero in this state and at power up; its value shall be maintained during a Hard Reset.		
8.3.3.2.3#6	"The Policy Engine shall transition to the PE_SRC_Negotiate_Capability state when: � A Request Message is received from the Sink."		
8.3.3.2.3#7	The Policy Engine shall transition to the PE_SRC_Discovery state when: · The Protocol Layer indicates that the message has not been sent and we are presently not connected.		
8.3.3.2.3#8	The Policy Engine shall transition to the PE_SRC_Hard_Reset state when: · The SenderResponseTimer times out. In this case a transition back to USB Default Operation is required. When: · The Port is not a Type-C Port · Or the Port is a Type-C Port and the Port Partners have not been PD Connected (the Type-C Port remains attached to a Port it has not had a PD Connection with during this attachment) · And the NoResponseTimer times out · And the HardResetCounter > nHardResetCount The Policy Engine shall do one of the following: · Transition to the PE_SRC_Discovery state. · Transition to the PE_SRC_Disabled state.		
8.3.3.2.3#10	The Policy Engine shall go to the ErrorRecovery state when: · This is a Type_C Port. · And the NoResponseTimer times out. · And the HardResetCounter > nHardResetCount.		
8.3.3.2.4 PE_SRC_Negotiate Capability			
8.3.3.2.4#1	On entry to the PE_SRC_Negotiate Capability state the Policy Engine shall ask the Device Policy Manager to evaluate the Request from the attached Sink.		
8.3.3.2.4#2	The response from the Device Policy Manager shall be one of the following: · The Request can be met. · The Request cannot be met · The Request could be met later from the Power reserve.		
8.3.3.2.4#3	The Policy Engine shall transition to the PE_SRC_Transition Supply state when:		

R2.0 Assertion #	Description	Test	Comments
	· The Request can be met.		
8.3.3.2.4#4	The Policy Engine shall transition to the PE_SRC_Capability Response state when: · The Request cannot be met. · Or the Request can be met later from the Power reserve.		
8.3.3.2.5 PE_SRC_Transition Supply			
8.3.3.2.5#1	The Policy Engine shall be in the PE_SRC_Transition Supply state while the Power Supply is transitioning from one power to another.		
8.3.3.2.5#2	On entry to the PE_SRC_Transition Supply state, the Policy Engine shall initialize and run the SourceActivityTimer (see Section 8.3.3.6 for details of Ping messaging for Source Ports), request the Protocol Layer to either send a GotoMin message (if this was requested by the Device Policy Manager) or otherwise an Accept message, wait for tSrcTransition, and inform the Device Policy Manager that it shall transition the Power Supply to the Requested power level.		
8.3.3.2.5#3	On exit from the PE_SRC_Transition Supply state the Policy Engine shall request the Protocol Layer to send a PS_RDY message.		
8.3.3.2.5#4	The Policy Engine shall transition to the PE_SRC_Ready state when: · The Device Policy Manager informs the Policy Engine that the Power Supply is ready.		
8.3.3.2.6 PE_SRC_Ready			
8.3.3.2.6#1	In the PE_SRC_Ready state the PD Source shall operate at a stable power with no ongoing negotiation.		
8.3.3.2.6#2	It shall respond to requests from the Sink, events from the Device Policy Manager and shall send out Ping messages to maintain the PD link if required.		
8.3.3.2.6#3	On entry to the PE_SRC_Ready state the Policy Engine shall initialize and run the SourceActivityTimer (see Section 8.3.3.6 for details of Ping messaging for Source Ports).		
8.3.3.2.6#4	If this is a DFP which needs to establish communication with a Cable Plug, the DFP shall initialize and run the DiscoverIdentityTimer (no GoodCRC message response yet received to Discover Identity message).		
8.3.3.2.6#5	The Policy Engine shall transition to the PE_SRC_Send_Capabilities state when: · The Device Policy Manager indicates that Source Capabilities have changed.		
8.3.3.2.6#6	The Policy Engine shall transition to the PE_SRC_Transition supply state when: · A GotoMin request is received from the Device Policy Manager for the attached Device to go to minimum power.		
8.3.3.2.6#7	The Policy Engine shall transition to the PE_SRC_Give Source Cap state when: · A Get_Source_Cap message is received from the Sink.		
8.3.3.2.6#8	The Policy Engine shall transition to the PE_SRC_Get Sink Cap state when: · The Device Policy Manager asks for the Sink's capabilities.		
8.3.3.2.8 PE_SRC_Capability Response			
8.3.3.2.8#1	The Policy Engine shall enter the PE_SRC_Capability Response state if there is a Request received from the Sink that cannot be met based on the present capabilities.		
8.3.3.2.8#2	On entry to the PE_SRC_Capability Response state the Policy Engine shall request the Protocol Layer to send one of the		

R2.0 Assertion #	Description	Test	Comments
	following: · Reject message – if the request cannot be met or the present Contract is invalid. · Wait message – if the request could be met later from the reserve.		
8.3.3.2.8#3	A Wait message shall not be sent if the present Conctract is Invalid.		
8.3.3.2.8#4	The Policy Engine shall transition to the PE_SRC_Ready state when: - There is an Explicit Contract and · The Reject message or Wait message has been sent and the present Contract is still valid or - A Wait message has been sent.		
8.3.3.2.8#5	The Policy Engine shall transition to the PE_SRC_Hard Reset state when: - There is an Explicit Contract and · The Reject message has been sent and the present Contract is Invalid (i.e. the Sink had to request a new value so instead we will return to default operation).		
8.3.3.2.8#6	The Policy Engine shall transition to the PE_SRC_Wait_New_Capabilities state when: · There is no Explicit Contract and · A Reject Message has been sent or · A Wait Message has been sent.		
8.3.3.2.9 PE_SRC_Hard Reset			
8.3.3.2.9#1	The Policy Engine shall transition from any state to the PE_SRC_Hard Reset state when · The NoResponseTimer times out and the HardResetCounter ≤ nHardResetCount or · The Device Policy Manager requests a Hard Reset.		
8.3.3.2.9#2	On entry to the PE_SRC_Hard Reset state the Policy Engine shall request the generation of Hard Reset signaling by the PHY Layer, and increment the HardResetCounter.		
8.3.3.2.9#3	Note that the NoResponseTimer shall continue to run in every state until it is stopped or times out.		
8.3.3.2.9#4	The Policy Engine shall transition to the PE_SRC_Transition to default state when: · The Hard Reset is complete.		
8.3.3.2.10 PE_SRC_Hard_Reset_Received			
8.3.3.2.10#1	The Policy Engine shall transition from any state to the PE_SRC_Hard_Reset_Received state when: · Hard Reset Signaling is detected.		
8.3.3.2.10#2	On entry to the PE_SRC_Hard_Reset_Received state the Policy Engine shall initialize and run the PSHardResetTimer.		
8.3.3.2.10#3	The Policy Engine shall transition to the PE_SRC_Transition_to_default state when: · The PSHardResetTimer times out.		
8.3.3.2.11 PE_SRC_Transition to default			
8.3.3.2.11#1	On entry to the PE_SRC_Transition to default state the Policy Engine shall - Indicate to the Device Policy Manager that the Power Supply shall Hard Reset (See Section 7.1.6) - request a reset of the local hardware. - for a type-C connector shall request the Device Policy Manager to set the Port Data Role to DFP and turn off Vconn.		
8.3.3.2.11#2	On exit from the PE_SRC_Transition to default state the Policy Engine shall: - for a Type-C connector shall request the Device Policy Manager to turn on Vconn. - initialize and run the NoResponseTimer.		

R2.0 Assertion #	Description	Test	Comments
	- inform the Protocol Layer that the Hard Reset is complete.		
8.3.3.2.11#3	Note that the NoResponseTimer shall continue to run in every state until it is stopped or times out.		
8.3.3.2.11#4	The Policy Engine shall transition to the PE_SRC_Startup state when: · The Device Policy Manager indicates that the Power Supply has reached the default level.		
8.3.3.2.12 PE_SRC_Give Source Cap			
8.3.3.2.12#1	On entry to the PE_SRC_Give Source Cap state the Policy Engine shall request the Device Policy Manager for the current system capabilities.		
8.3.3.2.12#2	The Policy Engine shall then request the Protocol Layer to send a Source Capabilities message containing these capabilities.		
8.3.3.2.12#3	The Policy Engine shall transition to the PE_SRC_Ready state when: · The Source Capabilities message has been successfully sent.		
8.3.3.2.13 PE_SRC_Get Sink Cap			
8.3.3.2.13#1	In this state the Policy Engine, due to a request from the Device Policy Manager, shall request the capabilities from the attached Sink.		
8.3.3.2.13#2	On entry to the PE_SRC_Get Sink Cap state the Policy Engine shall request the Protocol Layer to send a Get_Sink_Cap message in order to retrieve the Sink's capabilities.		
8.3.3.2.13#3	The Policy Engine shall then start the SenderResponseTimer.		
8.3.3.2.13#4	On exit from the PE_SRC_Get Sink Cap state the Policy Engine shall inform the Device Policy Manager of the outcome (capabilities, or response timeout).		
8.3.3.2.13#5	The Policy Engine shall transition~~s~~ to the PE_SRC_Ready state when: · A Sink Capabilities message is received. · Or SenderResponseTimer times out.		
8.3.3.2.14 PE_SRC_Wait_New_Capabilities			
8.3.3.2.14#1	The Policy Engine shall transition to the PE_SRC_Send_Capabilities state when: · The Device Policy Manager indicates that Source Capabilities have changed.		
8.3.3 Policy Engine Sink Port State Diagram			
8.3.3.3#1	The SinkActivityTimer shall not be run when operating at vSafe5V or when two systems using the Type_C connector are communicating, since Ping messages are optional.		
8.3.3.3.1 PE_SNK_Startup			
8.3.3.3.1#1	PE_SNK_Startup shall be the starting state for a Sink Policy Engine either on power up or after a Hard Reset		
8.3.3.3.1#2	On entry to this state the Policy Engine shall reset the Protocol Layer.		
8.3.3.3.1#3	Once the reset process completes, the Policy Engine shall transition to the PE_SNK_Discovery state for a Consumer only		
8.3.3.3.1#4	Once the reset process completes, the Policy Engine shall transition to the... to the PE_DB_CP_Check_for_VBUS state for a Type_B Consumer/Provider.		
8.3.3.3.2 PE_SNK_Discovery			
8.3.3.3.2#1	The Policy Engine shall transition to the PE_SNK_Wait for Capabilities state when: · The Device Policy Manager indicates that VBUS has been detected.		
8.3.3.3.2#2	The Policy Engine shall transition to the ErrorRecovery state when:		

R2.0 Assertion #	Description	Test	Comments
	· The Port is a Type-C connector and · The Port Partners have previously been PD Connected (the Type-C Port remains attached to a Port it has had a PD Connection with during this attachment) and · There has been a NoResponseTimer timeout and · The HardResetCounter > nHardResetCount.		
8.3.3.3.2#3	ThePolicy Engine shall transition to the PE_SNK_Hard_Reset state when: · The Port is not a Type-C connector and · There has been a NoResponseTimer timeout and · The HardResetCounter ≤ nHardResetCount.		
8.3.3.3.3 PE_SNK_Wait for Capabilities			
8.3.3.3.3#1	On entry to the PE_SNK_Wait for capabilities state the Policy Engine shall initialize and start the SinkWaitCapTimer.		
8.3.3.3.3#2	The Policy Engine shall transition to the PE_SNK_Evaluate Capability state when: · Source capabilities message is received.		
8.3.3.3.3#3	The Policy Engine shall transition to [USBType_C1.0] error recovery when: · This is a Type_C connector and · There has been a NoResponseTimer timeout and · The HardResetCounter > nHardResetCount.		
8.3.3.3.4 PE_SNK_Evaluate Capability			
8.3.3.3.4#1	On entry to the PE_SNK_Evaluate Capability state the Policy Engine shall request the Device Policy Manager to evaluate the supplied Source capabilities based on Local Policy.		
8.3.3.3.4#2	The Device Policy Manager shall indicate to the Policy Engine which new power level is required: _A selection from the present offered capabilities is to be made. _ Capability mismatch; offered power does not meet the device's requirements.		
8.3.3.3.4#3	The Policy Engine shall transition to the PE_SNK_Select Capability state when: · A response is received from the Device Policy Manager.		
8.3.3.3.5 PE_SNK_Select Capability			
8.3.3.3.5#1	On entry to the PE_SNK_Select Capability state the Policy Engine shall request the Protocol Layer to send a response message, based on the evaluation from the Device Policy Manager.		
8.3.3.3.5#2	The message shall be one of the following: · A Request from the offered Source Capabilities. · A Request from the offered Source Capabilities with an indication that another power level would be preferred.		
8.3.3.3.5#3	The Policy Engine shall initialize and run the SenderResponseTimer.		
8.3.3.3.5#4	The Policy Engine shall transition to the PE_SNK_Transition Sink state when: · An Accept message is received from the Source.		
8.3.3.3.5#5	The Policy Engine shall transition to the PE_SNK_Wait_for_Capabilities state when: · There is no Explicit Contract in place and · A Reject Message is received from the Source or · A Wait Message is received from the Source.		
8.3.3.3.5#6	The Policy Engine shall transition to the PE_SNK_Ready state when: · There is an Explicit Contract in place and · A Reject Message is received from the Source or · A Wait Message is received from the Source.		

R2.0 Assertion #	Description	Test	Comments
8.3.3.5#7	The Policy Engine shall transition to the PE_SNK_Hard Reset state when: · A SenderResponseTimer timeout occurs.		
8.3.3.3.6 PE_SNK_Transition Sink			
8.3.3.3.6#1	On entry to the PE_SNK_Transition Sink state the Policy Engine shall initialize and run the PSTransitionTimer (timeout will lead to a Hard Reset see Section 8.3.3.3.8) and shall then request the Device Policy Manager to transition the Sink's Power Supply to the new power level.		
8.3.3.3.6#1a	On exit from the PE_SNK_Transition Sink state the Policy Engine shall request the Device Policy Manager to transition the Sink's Power Supply to the new power level.		
8.3.3.3.6#2	The Policy Engine shall transition to the PE_SNK_Ready state when: · A PS_RDY message is received from the Source.		
8.3.3.3.7 PE_SNK_Ready			
8.3.3.3.7#1	In the PE_SNK_Ready state the PD Sink shall be operating at a stable power level with no ongoing negotiation.		
8.3.3.3.7#2	It shall respond to requests from the Source, events from the Device Policy Manager and may monitor for Ping messages to maintain the PD link.		
8.3.3.3.7#3	On entry to the PE_SNK_Ready state the Policy Engine shall do the following: · Initialize and run the SinkActivityTimer (except if in default operation see Section 6.5.4.2.).		
8.3.3.3.7#4	"If this is a DFP which needs to establish communication with a Cable Plug, the DFP shall: Initialize and run the DiscoverIdentityTimer (no GoodCRC message response yet received to Discover Identity message)."		
8.3.3.3.7#5	"The Policy Engine shall transition to the PE_SNK_Evaluate_Capability state when: A Source_Capabilities Message is received."		
8.3.3.3.7#6	The Policy Engine shall transition to the PE_SNK_Select Capability state when: · A new power level is requested by the Device Policy Manager. · A SinkRequestTimer timeout occurs.		
8.3.3.3.7#7	The Policy Engine shall transition to the PE_SNK_Transition Sink state when: · A GotoMin message is received.		
8.3.3.3.7#8	The Policy Engine shall transition to the PE_SNK_Hard Reset state when: · A SinkActivityTimer timeout occurs.		
8.3.3.3.7#9	The Policy Engine shall transition back to the PE_SNK_Ready state when: · A Ping message is received. Note this should not cause the SinkRequestTimer to be reinitialized.		
8.3.3.3.7#10	The Policy Engine shall transition to the PE_SNK_Give Sink Cap state when: · A Get_Sink_Cap message is received from the Protocol Layer.		
8.3.3.3.7#11	The Policy Engine shall transition to the PE_SNK_Get Source Cap state when: · The Device Policy Manager requests an update of the remote Source's capabilities.		
8.3.3.3.8 PE_SNK_Hard Reset			
8.3.3.3.8#1	The Policy Engine shall transition to the PE_SNK_Hard Reset state from any state when:		

R2.0 Assertion #	Description	Test	Comments
	· ((SinkWaitCapTimer timeout | · SinkActivityTimer timeout | · PSTransitionTimer timeout | · NoResponseTimer timeout) & · (HardResetCounter ≤ nHardResetCount)) | · Hard Reset request from Device Policy Manager		
8.3.3.3.8#2	Note: if the NoResponseTimer times out and the HardResetCounter is greater than nHardResetCount the Sink shall assume that the Source is non_responsive.		
8.3.3.3.8#3	On entry to the PE_SNK_Hard Reset state the Policy Engine shall request the generation of Hard Reset signaling by the PHY Layer and increment the HardResetCounter.		
8.3.3.3.8#4	The Policy Engine shall transition to the PE_SNK_Transition to default state when: · The Hard Reset is complete.		
8.3.3.3.9 PE_SNK_Transition to default			
8.3.3.3.9#1	The Policy Engine shall transition from any state to PE_SNK_Transition to default state when: · Hard Reset signaling is detected.		
8.3.3.3.9#2	When Hard Reset signaling is received or transmitted then the Policy Engine shall transition from any state to PE_SNK_Transition to default.		
8.3.3.3.9#3	On entry to the PE_SNK_Transition to default state the Policy Engine shall: - Indicate to the Device Policy Manager that the Sink shall transition to default. - request a reset of the local hardware - for a Type-C connector shall request that the Port Data Role is set to UFP.		
8.3.3.3.9#4	On exit from the PE_SNK_Transition to default state the Policy Engine shall initialize and run the NoResponseTimer and inform the Protocol Layer that the Hard Reset is complete.		
8.3.3.3.9#5	Note that the NoResponseTimer shall continue to run in every state until it is stopped or times out.		
8.3.3.3.9#6	The Policy Engine shall transition to the PE_SNK_Startup state when: · The Device Policy Manager indicates that the Sink has reached the default level.		
8.3.3.3.10 PE_SNK_Give Sink Cap			
8.3.3.3.10#1	On entry to the PE_SNK_Give Sink Cap state the Policy Engine shall request the Device Policy Manager for the current system capabilities.		
8.3.3.3.10#2	The Policy Engine shall then request the Protocol Layer to send a Sink Capabilities message containing these capabilities.		
8.3.3.3.10#3	The Policy Engine shall transition to the PE_SNK_Ready state when: · The Sink Capabilities message has been successfully sent.		
8.3.3.3.11 PE_SNK_Get Source Cap			
8.3.3.3.11#a	In the PE_SNK_Get_Source_Cap state the Policy Engine, due to a request from the Device Policy Manager, shall request the capabilities from the attached Source.		
8.3.3.3.11#1	On entry to the PE_SNK_Get Source Cap state the Policy Engine shall request the Protocol Layer to send a Get_Source_Cap message in order to retrieve the Source's capabilities.		
8.3.3.3.11#2	The Policy Engine shall transition to the PE_SNK_Ready state when:		

R2.0 Assertion #	Description	Test	Comments
	· The Get_Source_Cap message is sent.		
8.3.3.4.1.1 PE_SRC_Send Soft Reset			
8.3.3.4.1.1#1	The PE_SRC_Send Soft Reset state shall be entered from any state when a Protocol Error is detected by the Protocol Layer		
8.3.3.4.1.1#2	The PE_SRC_Send Soft Reset state shall be entered from any state when...a message has not been sent after retries.		
8.3.3.4.1.1#3	On entry to the PE_SRC_Send Soft Reset state the Policy Engine shall request the Protocol Layer to perform a Soft Reset then shall send a Soft Reset message to the Sink and initialize and run the SenderResponseTimer.		
8.3.3.4.1.1#4	The Policy Engine shall transition to the PE_SRC_Send Capabilities (Source) state when: • An Accept message has been received.		
8.3.3.4.1.1#5	The Policy Engine shall transition to the PE_SRC_Hard Reset or PE_SNK_Hard_Reset state depending on its default role as either a Source or Sink Port (see Section 6.7.2) when: • A SenderResponseTimer timeout occurs. _Or the Protocol Layer indicates that a transmission error has occurred.		
8.3.3.4.1.1#6	The decision as to whether to go to PE_SRC_Hard Reset or PE_SNK_Hard Reset shall depend on the type of device: • The Source port in a Provider or Provider/Consumer shall go to PE_SRC_Hard Reset.		
8.3.3.4.1.1#7	The decision as to whether to go to PE_SRC_Hard Reset or PE_SNK_Hard Reset shall depend on the type of device: • The Source port in a Type-B Consumer/Provider shall go to PE_SRC_Hard Reset i.e. revert to default operation as Sink Port.		
8.3.3.4.1.2 PE_SRC_Soft Reset			
8.3.3.4.1.2#1	The PE_SRC_Soft Reset state shall be entered from any state when a Soft Reset message is received from the Protocol Layer.		
8.3.3.4.1.2#2	On entry to the PE_SRC_Soft Reset state the Policy Engine shall reset the Protocol Layer		
8.3.3.4.1.2#3	On entry to the PE_SRC_Soft Reset state the Policy Engine...shall then request the Protocol Layer to send an Accept message.		
8.3.3.4.1.2#4	The Policy Engine shall transition to the PE_SRC_Send Capabilities state (See Section 8.3.3.2.3) when: • The Accept message has been sent.		
8.3.3.4.1.2#5	The Policy Engine shall transition to the PE_SRC_Hard_Reset or PE_SNK_Hard_Reset state depending on its default role as either a Source or Sink Port (see Section 6.7.2) when: · The Protocol Layer indicates that a transmission error has occured.		
8.3.3.4.1.2#6	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: · The Source Port in a Provider or Provider/Consumer shall go to PE_SRC_Hard_Reset.		
8.3.3.4.1.2#7	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: · The Source Port in a Consumer/Provider shall go to PE_SNK_Hard_Reset i.e. revert to default operation as Sink Port.		
8.3.3.4.2.1 PE_SNK_Send Soft Reset			
8.3.3.4.2.1#1	The PE_SNK_Send Soft Reset state shall be entered from any state when a Protocol Error is detected by the Protocol Layer		
8.3.3.4.2.1#2	The PE_SNK_Send Soft Reset state shall be entered from any state...when a message has not been sent after retries.		

R2.0 Assertion #	Description	Test	Comments
8.3.3.4.2.1#3	On entry to the PE_SNK_Send Soft Reset state the Policy Engine shall request the Protocol Layer to perform a Soft Reset then shall send a Soft Reset message to the Source and initialize and run the SenderResponseTimer.		
8.3.3.4.2.1#4	The Policy Engine shall transition to the PE_SNK_Wait for Capabilities state when: • An Accept message has been received.		
8.3.3.4.2.1#5	The Policy Engine shall transition to the PE_SRC_Hard_Reset or PE_SNK_Hard_Reset state when: · A SenderResponseTimer timeout occurs. · Or the Protocol Layer indicates that a transmission error has occured.		
8.3.3.4.2.1#6	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: The Sink Port in a Provider/Consumer shall go to PE_SRC_Hard_Reset.		
8.3.3.4.2.1#7	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: · The Sink Port in a Consumer or Consumer/Provider shall go to PE_SNK_Hard_Reset i.e. revert to default operation as Sink Port.		
8.3.3.4.2.2 PE_SNK_Soft Reset			
8.3.3.4.2.2#1	The PE_SNK_Soft Reset state shall be entered from any state when a Soft Reset message is received from the Protocol Layer.		
8.3.3.4.2.2#2	On entry to the PE_SNK_Soft Reset state the Policy Engine shall reset the Protocol Layer		
8.3.3.4.2.2#3	On entry to the PE_SNK_Soft Reset state the Policy Engine...shall then request the Protocol Layer to send an Accept message.		
8.3.3.4.2.2#4	The Policy Engine shall transition to the PE_SNK_Wait for Capabilities state when: • The Accept message has been sent.		
8.3.3.4.2.2#5	The Policy Engine shall transition to the PE_SRC_Hard_Reset or PE_SNK_Hard_Reset state depending on its default role as either a Source or Sink Port (see Section 6.7.2) when: · The Protocol Layer indicates that a transmission error has occured.		
8.3.3.4.2.2#6	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: · The Sink Port in a Type-A Provider/Consumer shall go to PE_SRC_Hard_Reset.		
8.3.3.4.2.2#7	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: · The Sink Port in a Consumer or Consumer/Provider shall go to PE_SNK_Hard_Reset i.e. revert to default operation as Sink Port.		
8.3.3.5.3.1 PE_SRC_Ping			
8.3.3.5.3.1#1	On entry to the PE_SRC_Ping state (from the PE_SRC_Transition Supply or PE_SRC_Ready states) the Policy Engine shall request the Protocol Layer to send a Ping message.		
8.3.3.5.3.1#2	The Policy Engine shall transition back to the previous state (PE_SRC_Transition Supply or PE_SRC_Ready) state (see Figure 8 39) when: • The Ping message has been successfully sent.		
8.3.3.5.3.1#3	On re_entry to the PE_SRC_Transition Supply or PE_SRC_Ready states the Policy Engine shall not perform any of the “Actions on Entry” except for initializing and running		

R2.0 Assertion #	Description	Test	Comments
	the SourceActivityTimer		
8.3.3.6 Dual_Role Port State Diagrams			
8.3.3.6#1	Dual_Role Ports that combine Source and Sink capabilities shall comprise Source and Sink Policy Engine state machines.		
8.3.3.6#2	In addition they shall have the capability to perform a Power Role Swap from the PE_SRC_Ready or PE_SNK_Ready states		
8.3.3.6#3	In addition they...shall return to default operation on a Hard Reset.		
8.3.3.6#4	Type_C Dual_Role Ports shall have, in addition, the capability to perform a Data role Swap from the PE_SRC_Ready or PE_SNK_Redy states and shall return to their default operation on a Hard Reset.		
8.3.3.6#5	The following sections describe how the Source and Sink state diagrams shall be combined in the case of Hard Reset, the role swap process and the operation in the case of a dead battery.		
8.3.3.6.1.1 PE_PRS_SRC_SNK_Ping			
8.3.3.6.1.1#1	The Policy Engine shall transition to the PE_PRS_SRC_SNK_Ping state from the PE_PRS_SRC_SNK_Transition to off state, due to a SourceActivityTimer timeout.		
8.3.3.6.1.1#1a	On entry to the PE_PRS_SRC_SNK_Ping state the Policy Engine shall request the Protocol Layer to send a Ping message.		
8.3.3.6.1.1#2	The Policy Engine shall transition back to the PE_PRS_SRC_SNK_Transition to off state (see Figure 8 52) when: • The Ping message has been successfully sent.		
8.3.3.6.1.1#3	The Policy Engine shall transition to the PE_SRC_Hard Reset state when: • The Ping message has not been sent after retries (a GoodCRC message has not been received). A soft reset shall not be initiated in this case.		
8.3.3.6.1.2.1 PE_PRS_SNK_SRC_Ping			
8.3.3.6.1.2.1#1	The Policy Engine shall transition to the PE_PRS_SNK_SRC_Ping state, from the PE_PRS_SNK_SRC_Source on state, due to a SourceActivityTimer timeout.		
8.3.3.6.1.2.1#2	On entry to the PE_PRS_SNK_SRC_Ping state from the PE_PRS_SNK_SRC_Source on state the Policy Engine shall request the Protocol Layer to send a Ping message.		
8.3.3.6.1.2.1#3	The Policy Engine shall transition back to the PE_PRS_SNK_SRC_Source on state (see Figure 8 53) when: • The Ping message has been successfully sent.		
8.3.3.6.1.2.1#4	The Policy Engine shall transition to the PE_SNK_Hard Reset state for a Sink Port when: • The Ping message has not been sent after retries (a GoodCRC message has not been received). A soft reset shall not be initiated in this case.		
8.3.3.6.1.3.1 PE_PC_SNK_Hard Reset			
8.3.3.6.1.3.1#1	The Policy Engine shall transition to the PE_PC_SNK_Hard Reset state for a Provider/Consumer Port in SInk Role from any state when: • ((SinkWaitCapTimer timeout | • SinkActivityTimer timeout | • PSTransitionTimer timeout | • NoResponseTimer timeout) & • (HardResetCounter ≤ nHardResetCount)) | • Hard Reset request from Device Policy Manager		

R2.0 Assertion #	Description	Test	Comments
8.3.3.6.1.3.1#2	The Policy Engine shall transition to the PE_PC_SNK_Swap_Recovery state when: · The Hard Reset is complete.		
8.3.3.6.1.3.2 PE_PC_SNK_Swap Recovery			
8.3.3.6.1.3.2#1	The Policy Engine shall transition to the PE_PC_SNK_Swap Recovery state from any state to when: • Hard Reset signaling is detected.		
8.3.3.6.1.3.2#2	On entry to the PE_PC_SNK_Swap Recovery state the Policy Engine shall initialize and run the SwapRecoveryTimer.		
8.3.3.6.1.3.2#3	The Policy Engine shall transition to the PE_SRC_Transition to Default state for a Source Port when: • The SwapRecoveryTimer times out.		
8.3.3.6.1.4.1 PE_CP_SRC_Hard Reset			
8.3.3.6.1.4.1#1	The Protocol Engine shall transition from any state to the PE_CP_SRC_Hard Reset state for a Consumer/Provider in a Source Role when • The NoResponseTimer times out and the HardResetCounter ≤ nHardResetCount or _ The Device Policy Manager requests a hard reset.		
8.3.3.6.1.4.1#2	The Policy Engine shall transition to the PE_CP_SRC_Transition_to_off state when: · The Hard Reset is complete.		
8.3.3.6.1.4.2 PE_CP_SRC_Transition to off			
8.3.3.6.1.4.2#1	The Policy Engine shall transition from any state to the PE_CP_SRC_Transition to off state for a Consumer/Provider in Source Role when: • Hard Reset signaling is detected.		
8.3.3.6.1.4.2#2	On Entry to the PE_CP_SRC_Transition to off state the Policy Engine shall tell the Device Policy Manager to turn off the Power Supply.		
8.3.3.6.1.4.2#3	The Policy Engine shall transition to the PE_SNK_Transition_to_default state when: _ The Power Supply has been turned off.		
8.3.3.6.1.5 BFSK Consumer/Provider Dead Battery/Power Loss State Diagram			
8.3.3.6.1.5#1	After the Consumer/Provider Policy Engine has transitioned to the PE_SRC_Send_Capabilities state, its subsequent state operation shall conform to that of a Consumer/Provider which has completed a Power Role Swap (see Section 8.3.3.6.3.2).		
8.3.3.6.1.5.1 PE_DB_CP_Check_for_VBUS state			
8.3.3.6.1.5.1#1	The Policy Engine for a Consumer/Provider shall initially start in the PE_SNK_Startup state.		
8.3.3.6.1.5.1#2	Once the Protocol Layer has been reset it shall transition to the PE_DB_CP_Check_for_VBUS state.		
8.3.3.6.1.5.1#3	On entry to the PE_DB_CP_Check_for_VBUS state the Policy Engine shall initialize and run the DBDetectTimer and stop the NoResponseTimer.		
8.3.3.6.1.5.1#4	The Policy Engine shall transition to the PE_SNK_Wait_for_Capabilities state when: · VBUS is greater than vSafe0V.		
8.3.3.6.1.5.1#5	The Policy Engine shall transition to the PE_DB_CP_Power_VBUS_DB state when: · The DBDetectTimer has timed out and · VBUS is within vSafe0V.		
8.3.3.6.1.5.2 PE_DB_CP_Power_VBUS_DB state			
8.3.3.6.1.5.2#1	On entry to the PE_DB_CP_Power_VBUS_DB state the Policy Engine shall tell the Device Policy Manager to apply vSafeDB to VBUS.		
8.3.3.6.1.5.2#2	The Policy Engine shall transition to the		

R2.0 Assertion #	Description	Test	Comments
	PE_DB_CP_Wait_For_Bit_Stream state when: · vSafeDB is on VBUS.		
8.3.3.6.1.5.3 PE_DB_CP_Wait_For_Bit_Stream state			
8.3.3.6.1.5.3#1	On entry to the PE_DB_CP_Wait_For_Bit_Stream state the Policy Engine shall initialize and run the BitStreamDetectTimer.		
8.3.3.6.1.5.3#2	The Policy Engine shall transition to the PE_DB_CP_Power_VBUS_5V state when: · The PHY Layer indicates that Bit Stream signaling has been received.		
8.3.3.6.1.5.3#3	The Policy Engine shall transition to the PE_DB_CP_Unpower_VBUS state when: · The BitStreamDetectTimer times out.		
8.3.3.6.1.5.4 PE_DB_CP_Power_VBUS_5V state			
8.3.3.6.1.5.4#1	On entry to the PE_DB_CP_Power_VBUS_5V state the Policy Engine shall request the Device Policy Manager to apply vSafe5V to VBUS.		
8.3.3.6.1.5.4#2	The Policy Engine shall transition to the PE_DB_CP_Wait_Bit_Stream_Stop state when: · vSafe5V is present on VBUS.		
8.3.3.6.1.5.5 PE_DB_CP_Wait_Bit_Stream_Stop state			
8.3.3.6.1.5.5#1	On entry to the PE_DB_CP_Wait_Bit_Stream_Stop state the Policy Engine shall initialize and run the DeviceReadyTimer.		
8.3.3.6.1.5.5#2	The Policy Engine shall transition to the PE_SRC_Send_Capabilities state when: · An indication is received from the PHY Layer that the Bit Stream has stopped.		
8.3.3.6.1.5.5#3	The Policy Engine shall transition to the PE_DB_CP_Unpower_VBUS state when: · The DeviceReadyTimer times out.		
8.3.3.6.1.5.6 PE_DB_CP_Unpower_VBUS state			
8.3.3.6.1.5.6#1	On entry to the PE_DB_CP_Unpower_VBUS state the Policy Engine shall tell the Device Policy Manager to apply vSafe0V to VBUS.		
8.3.3.6.1.5.6#2	The Policy Engine shall transition to the PE_DB_CP_PS_Discharge state when: · vSafe0V has been applied to VBUS.		
8.3.3.6.1.5.7 PE_DB_CP_PS_Discharge state			
8.3.3.6.1.5.7#1	On entry to the PE_DB_CP_PS_Discharge state the Policy Engine shall initialize and run the DBSourceOffTimer.		
8.3.3.6.1.5.7#2	The Policy Engine shall transition to the PE_DB_CP_Check_for_VBUS state when: · The DBSourceOffTimer times out.		
8.3.3.6.1.6.1 PE_DB_PC_Unpowered state			
8.3.3.6.1.6.1#1	The PE_DB_PC_Unpowered state shall be entered from any state when: · VBUS is within vSafe0V and · The Provider/Consumer is powered off or has insufficient power to operate.		
8.3.3.6.1.6.1#2	On entry to the PE_DB_PC_Unpowered state the Policy Engine shall wait for vSafeDB to appear on VBUS in order to start the Dead Battery detection process.		
8.3.3.6.1.6.1#3	If a Bit Stream is currently being transmitted then this shall be stopped within tBitStreamOff of vSafe0V appearing on VBUS.		
8.3.3.6.1.6.1#4	The Policy Engine shall transition to the PE_DB_PC_Check_Power state when: · vSafeDB is present on VBUS,.or		
8.3.3.6.1.6.1#5	The Policy Engine shall transition to the		

R2.0 Assertion #	Description	Test	Comments
	PE_DB_PC_Check_Power state when: · The Provider/Consumer is ready to power VBUS		
8.3.3.6.1.6.2 PE_DB_PC_Check_Power state			
8.3.3.6.1.6.2#1	On entry to the PE_DB_PC_Check_Power state the Policy Engine shall decide whether it is able and willing to supply power to the Port.		
8.3.3.6.1.6.2#2	The Policy Engine shall transition to the PE_SRC_Transition_to_default state when: · It is willing to power VBUS.		
8.3.3.6.1.6.2#3	The Policy Engine shall transition to the PE_DB_PC_Send_Bit_Stream state when: · It wants to be powered by the Consumer/Provider.		
8.3.3.6.1.6.2#4	The Policy Engine shall stay in the PE_DB_PC_Check_Power state when: · The Provider/Consumer does not want to either power the Port or be powered.		
8.3.3.6.1.6.3 PE_DB_PC_Send_Bit_Stream state			
8.3.3.6.1.6.3#1	On entry to the PE_DB_PC_Send_Bit_Stream state the Policy Engine shall request the PHY Layer to start sending the Bit Stream (see Section 4.1.1).		
8.3.3.6.1.6.3#2	The Policy Engine shall transition to the PE_DB_PC_Wait_to_Detect state when: · The Bit Stream is being sent.		
8.3.3.6.1.6.4 PE_DB_PC_Wait_to_Detect state			
8.3.3.6.1.6.4#1	On entry to the PE_DB_PC_Wait_to_Detect state the Policy Engine shall initialize and run the WaitForPowerTimer to allow the Consumer/Provider time to detect the Bit Stream and apply power to VBUS.		
8.3.3.6.1.6.4#2	The Policy Engine shall transition to the PE_DB_PC_Wait_to_Start state when: · The DeadBatteryTimer times out.		
8.3.3.6.1.6.5 PE_DB_PC_Wait_to_Start state			
8.3.3.6.1.6.5#1	On entry to the PE_DB_PC_Wait_to_Start state the Policy Engine shall wait until it is ready to negotiate Capabilities.		
8.3.3.6.1.6.5#2	On exit from the PE_DB_PC_Wait_to_Start state the Policy Engine shall request the PHY Layer to stop sending the Bit Stream.		
8.3.3.6.1.6.5#3	The Policy Engine shall transition to the PE_SNK_Wait_for_Capabilities state when: · The Policy Engine is ready to negotiate Capabilities.		
8.3.3.6.2.1 Type_C DRP Policy Engine in DFP to UFP Data Role Swap State Diagram			
8.3.3.6.2.1#1	Figure 8_50 shows the additional state diagram required to perform a Data Role Swap from Type_C DFP to UFP operation and the changes that shall be followed for error and Hard Reset handling.		
8.3.3.6.2.1.1 PE_SRC_Ready or PE_SNK_Ready			
8.3.3.6.2.1.1#1	The Data Role Swap process shall start only from either the PE_SRC_Ready or PE_SNK_Ready state where power is stable.		
8.3.3.6.2.1.1#2	The Policy Engine shall transition to the PE_DRS_DFP_UFP_Evaluate_DR_Swap state when: · A DR_Swap message is received.		
8.3.3.6.2.1.1#3	The Policy Engine shall transition to either the PE_SRC_Hard_Reset or PE_SNK_Hard_Reset states when: · A DR_Swap Message is received and · There are one or more Active Modes (Modal Operation).		
8.3.3.6.2.1.1#4	The Policy Engine shall transition to the PE_DRS_DFP_UFP_Send_DR_Swap state when: · The Device Policy Manager indicates that a Data Role Swap		

R2.0 Assertion #	Description	Test	Comments
	is required.		
8.3.3.6.2.1.2 PE_DRS_DFP_UFP_Evaluate_DR_Swap state			
8.3.3.6.2.1.2#1	On entry to the PE_DRS_DFP_UFP_Evaluate_DR_Swap state the Policy Engine shall ask the Device Policy Manager whether a Data Role Swap can be made.		
8.3.3.6.2.1.2#2	The Policy Engine shall transition to the PE_DRS_DFP_UFP_Accept_DR_Swap state when: · The Device Policy Manager indicates that a Data Role Swap is ok.		
8.3.3.6.2.1.2#3	The Policy Engine shall transition to the PE_DRS_DFP_UFP_Reject_DR_Swap state when: · The Device Policy Manager indicates that a Data Role Swap is not ok.		
8.3.3.6.2.1.2#4	The Policy Engine shall transition to the PE_DRS_DFP_UFP_Reject_DR_Swap state when: · Or further evalution of the Data Role Swap request is needed.		
8.3.3.6.2.1.3 PE_DRS_DFP_UFP_Accept_DR_Swap state			
8.3.3.6.2.1.3#1	On entry to the PE_DRS_DFP_UFP_Accept_DR_Swap state the Policy Engine shall request the Protocol Layer to send an Accept message.		
8.3.3.6.2.1.3#2	The Policy Engine shall transition to the PE_DRS_DFP_UFP_Change_to_UFP state when: · The Accept message has been sent.		
8.3.3.6.2.1.4 PE_DRS_DFP_UFP_Change_to_UFP state			
8.3.3.6.2.1.4#1	On entry to the PE_DRS_DFP_UFP_Change_to_UFP state the Policy Engine shall request the Device Policy Manager to change the Port from a DFP to a UFP.		
8.3.3.6.2.1.4#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state when: · The Device Policy Manager indicates that the Type_C Port has been changed to a UFP.		
8.3.3.6.2.1.5 PE_DRS_DFP_UFP_Send_DR_Swap state			
8.3.3.6.2.1.5#1	On entry to the PE_DRS_DFP_UFP_Send_DR_Swap state the Policy Engine shall request the Protocol Layer to send a DR_Swap message and shall start the SenderResponseTimer.		
8.3.3.6.2.1.5#2	On exit from the PE_DRS_DFP_UFP_Send_DR_Swap state the Policy Engine shall stop the SenderResponseTimer.		
8.3.3.6.2.1.5#3	The Policy Engine shall continue as a DFP and shall transition to either the PE_SRC_Ready or PE_SNK_Ready state when: · A Reject message is received. · Or a Wait message is received. · Or the SenderResponseTimer times out.		
8.3.3.6.2.1.5#4	The Policy Engine shall transition to the PE_DRS_DFP_UFP_Change_to_UFP state when: · An Accept message is received.		
8.3.3.6.2.1.6 PE_DRS_DFP_UFP_Reject_DR_Swap state			
8.3.3.6.2.1.6#1	On entry to the PE_DRS_DFP_UFP_Reject_DR_Swap state the Policy Engine shall request the Protocol Layer to send: · A Reject message if the device is unable to perform a Data Role Swap at this time.		
8.3.3.6.2.1.6#2	On entry to the PE_DRS_DFP_UFP_Reject_DR_Swap state the Policy Engine shall request the Protocol Layer to send: · A Wait message if further evaluation of the Data Role Swap request is required.		
8.3.3.6.2.1.6#3	The Policy Engine shall continue as a DFP and shall transition to either the PE_SRC_Ready or PE_SNK_Ready state when: · The Reject or Wait message has been sent.		
8.3.3.6.2.1.7 Type_C error recovery			

R2.0 Assertion #	Description	Test	Comments
8.3.3.6.2.1.7#1	The Policy Engine shall transition to ErrorRecovery when: · A protocol error is detected by the Protocol Layer or		
8.3.3.6.2.1.7#2	The Policy Engine shall transition to ErrorRecovery when: · A message has not been sent after retries or		
8.3.3.6.2.1.7#3	The Policy Engine shall transition to ErrorRecovery when: · Hard Reset signaling is received.		
8.3.3.6.2.2 Type_c DRP Policy Engine in UFP to DFP Data Role Swap State Diagram			
8.3.3.6.2.2#1	Figure 8-50 shows the additional state diagram required to perform a Data Role Swap from Type-C DRP UFP to DFP operation and the changes that shall be followed for error and Hard Reset handling.		
8.3.3.6.2.2.1 PE_SRC_Ready or PE_SNK_Ready state			
8.3.3.6.8.1#1	The Data Role Swap process shall start only from the either the PE_SRC_Ready or PE_SNK_Ready state where power is stable.		
8.3.3.6.8.1#2	The Policy Engine shall transition to the PE_DRS_UFP_DFP_Evaluate_DR_Swap state when: · A DR_Swap message is received.		
8.3.3.6.8.1#3	The Policy Engine shall transition to either the PE_SRC_Hard_Reset or PE_SNK_Hard_Reset states when: · A DR_Swap Message 1 is received and · There are one or more Active Modes (Modal Operation).		
8.3.3.6.8.1#4	The Policy Engine shall transition to the PE_DRS_UFP_DFP_Send_DR_Swap state when: · The Device Policy Manager indicates that a Data Role Swap is required.		
8.3.3.6.2.2.2 PE_DRS_UFP_DFP_Evaluate_DR_Swap state			
8.3.3.6.2.2.2#1	On entry to the PE_DRS_UFP_DFP_Evaluate_DR_Swap state the Policy Engine shall ask the Device Policy Manager whether a Data Role Swap can be made.		
8.3.3.6.2.2.2#2	The Policy Engine shall transition to the PE_DRS_UFP_DFP_Accept_DR_Swap state when: · The Device Policy Manager indicates that a Data Role Swap is ok.		
8.3.3.6.2.2.2#3	The Policy Engine shall transition to the PE_DRS_UFP_DFP_Reject_DR_Swap state when: · The Device Policy Manager indicates that a Data Role Swap is not ok. · Or further evalution of the Data Role Swap request is needed.		
8.3.3.6.2.2.3 PE_DRS_UFP_DFP_Accept_DR_Swap state			
8.3.3.6.2.2.3#1	On entry to the PE_DRS_UFP_DFP_Accept_DR_Swap state the Policy Engine shall request the Protocol Layer to send an Accept message.		
8.3.3.6.2.2.3#2	The Policy Engine shall transition to the PE_DRS_UFP_DFP_Change_to_DFP state when: · The Accept message has been sent.		
8.3.3.6.2.2.4 PE_DRS_UFP_DFP_Change_to_DFP state			
8.3.3.6.2.2.4#1	On entry to the PE_DRS_UFP_DFP_Change_to_DFP state the Policy Engine shall request the Device Policy Manager to change the Port from a UFP to a DFP.		
8.3.3.6.8.4#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state when: · The Device Policy Manager indicates that the Type_C Port has been changed to a DFP.		
8.3.3.6.2.2.5 PE_DRS_UFP_DFP_Send_DR_Swap state			
8.3.3.6.2.2.5#1	On entry to the PE_DRS_UFP_DFP_Send_DR_Swap state the Policy Engine shall request the Protocol Layer to send a DR_Swap message and shall start the SenderResponseTimer.		

R2.0 Assertion #	Description	Test	Comments
8.3.3.6.2.2.5#2	On exit from the PE_DRS_UFP_DFP_Send_DR_Swap state the Policy Engine shall stop the SenderResponseTimer.		
8.3.3.6.2.2.5#3	The Policy Engine shall continue as a UFP and shall transition to either the PE_SRC_Ready or PE_SNK_Ready state when: · A Reject message is received. · Or a Wait message is received. · Or the SenderResponseTimer times out.		
8.3.3.6.2.2.5#4	The Policy Engine shall transition to the PE_DRS_UFP_DFP_Change_to_DFP state when: · An Accept message is received.		
8.3.3.6.2.2.6 PE_DRS_UFP_DFP_Reject_DR_Swap state			
8.3.3.6.2.2.6#1	On entry to the PE_DRS_UFP_DFP_Reject_DR_Swap state the Policy Engine shall request the Protocol Layer to send: · A Reject message if the device is unable to perform a Data Role Swap at this time.		
8.3.3.6.2.2.6#2	On entry to the PE_DRS_UFP_DFP_Reject_DR_Swap state the Policy Engine shall request the Protocol Layer to send: · A Wait message if further evaluation of the Data Role Swap request is required.		
8.3.3.6.2.2.6#3	The Policy Engine shall continue as a UFP and shall transition to the either the PE_SRC_Ready or PE_SNK_Ready state when: · The Reject or Wait message has been sent.		
8.3.3.6.3.1 Policy Engine in Source to Sink Power Role Swap State Diagram			
8.3.3.6.3.1#1	Dual_Role Ports that combine Source and Sink capabilities shall comprise Source and Sink Policy Engine state machines.		
8.3.3.6.5#2	In addition they shall have the capability to do a Power Role Swap from the PE_SRC_Ready state and shall return to default operation on a Hard Reset.		
8.3.3.6.5#3	Figure 8_52 shows the additional state diagram required to perform a Power Role Swap from Source to Sink roles and the changes that shall be followed for error and Hard Reset handling.		
8.3.3.6.3.1.1 PE_SRC_Ready			
8.3.3.6.3.1.1#1	The Power Role Swap process shall start only from the PE_SRC_Ready state where power is stable.		
8.3.3.6.3.1.1#2	The Policy Engine shall transition to the PE_PRS_SRC_SNK_Evaluate_PR_Swap state when: • A PR_Swap message is received.		
8.3.3.6.3.1.1#3	The Policy Engine shall transition to the PE_PRS_SRC_SNK_Send_PR_Swap state when: • The Device Policy Manager indicates that a Power Role Swap is required.		
8.3.3.6.3.1.2 PE_PRS_SRC_SNK_Evaluate_PR_Swap			
8.3.3.6.3.1.2#1	On entry to the PE_PRS_SRC_SNK_Evaluate_PR_Swap state the Policy Engine shall ask the Device Policy Manager whether a Swap can be made.		
8.3.3.6.3.1.2#2	The Policy Engine shall transition to the PE_PRS_SRC_SNK_Accept_PR_Swap state when: • The Device Policy Manager indicates that a Role Swap is ok.		
8.3.3.6.3.1.2#3	The Policy Engine shall transition to the PE_PRS_SRC_SNK_Reject_PR_Swap state when: • The Device Policy Manager indicates that a Role Swap is not ok.		
8.3.3.6.3.1.2#4	The Policy Engine shall transition to the PE_PRS_SRC_SNK_Reject_PR_Swap state when: • Or further evalution of the Role Swap request is needed.		
8.3.3.6.3.1.3 PE_PRS_SRC_SNK_Accept_PR_Swap			
8.3.3.6.3.1.3#1	On entry to the PE_PRS_SRC_SNK_Accept_PR_Swap state		

R2.0 Assertion #	Description	Test	Comments
	the Policy Engine shall request the Protocol Layer to send an Accept message.		
8.3.3.6.3.1.3#2	The Policy Engine shall transition to the PE_PRS_SRC_SNK_Transition to off state when: • The Accept message has been sent.		
8.3.3.6.3.1.4 PE_PRS_SRC_SNK_Transition to off			
8.3.3.6.3.1.4#1	On entry to the PE_PRS_SRC_SNK_Transition to off state the Policy Engine shall wait tSrcTransition and then request the Device Policy Manager to turn off the Source and shall initialize and run the SourceActivityTimer (see Section 8.3.3.6.1.1.1 for use of Ping messaging for Dual-Role Ports which are initially Source Ports).		
8.3.3.6.3.1.4#2	The Policy Engine shall transition to the PE_PRS_SRC_SNK_Source off state when: • The Device Policy Manager indicates that the Source has been turned off and · The Port is not a [USBType_C1.0] DRP		
8.3.3.6.3.1.4#3	The Policy Engine shall transition to the PE_PRS_SRC_SNK_Assert_Rd state when: · The Device Policy Manager indicates that the Source has been turned off and · The Port is a [USBType_C1.0] DRP		
8.3.3.6.3.1.5 PE_PRS_SRC_SNK_Assert_Rd			
8.3.3.6.3.1.5#1	On entry to the PE_PRS_SRC_SNK_Assert_Rd state the Policy Engine shall request the Device Policy Manager to change the resistor asserted on the CC wire from Rp to Rd.		
8.3.3.6.3.1.5#2	The Policy Engine shall transition to the PE_PRS_SRC_SNK_Source_off state when: · The Device Policy Manager indicates that Rd is asserted.		
8.3.3.6.3.1.6 PE_PRS_SRC_SNK_Source off			
8.3.3.6.3.1.6#1	On entry to the PE_PRS_SRC_SNK_Source off state the Policy Engine shall request the Protocol Layer to send a PS_RDY message		
8.3.3.6.3.1.6#2	On entry to the PE_PRS_SRC_SNK_Source off state the Policy Engine shall...shall start the PSSourceOnTimer.		
8.3.3.6.3.1.6#3	On exit from the PE_PRS_SRC_SNK_Source off state the Policy Engine shall stop the PSSourceOnTimer.		
8.3.3.6.3.1.6#4	The Policy Engine shall transition to the PE_SNK_Startup state when: • A PS_RDY message is received indicating that the remote Source is now supplying power.		
8.3.3.6.3.1.6#5	The Policy Engine shall transition to the PE_SRC_Hard_Reset or PE_SNK_Hard_Reset state depending on its default role as either a Source or Sink Port (see Section 6.7.2) when: · The PSSourceOnTimer times out or · The PS_RDY message is not sent after retries (a GoodCRC message has not been received). A soft reset shall not be initiated in this case.		
8.3.3.6.3.1.6#6	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: · The Source Port in a Provider or Provider/Consumer shall go to PE_SRC_Hard_Reset.		
8.3.3.6.3.1.6#7	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: · The Source Port in a Consumer/Provider shall go to PE_SNK_Hard_Reset i.e. revert to default operation as Sink Port.		
8.3.3.6.3.1.7 PE_PRS_SRC_SNK_Send Swap			
8.3.3.6.3.1.7#1	On entry to the PE_PRS_SRC_SNK_Send Swap state the		

R2.0 Assertion #	Description	Test	Comments
	Policy Engine shall request the Protocol Layer to send a Swap message		
8.3.3.6.3.1.7#2	On entry to the PE_PRS_SRC_SNK_Send Swap state the Policy Engine shall...shall start the SenderResponseTimer.		
8.3.3.6.3.1.7#3	On exit from the PE_PRS_SRC_SNK_Send Swap state the Policy Engine shall stop the SenderResponseTimer.		
8.3.3.6.3.1.7#4	The Policy Engine shall transition to the PE_SRC_Ready state for a Source Port when: • A Reject message is received.		
8.3.3.6.3.1.7#5	The Policy Engine shall transition to the PE_SRC_Ready state for a Source Port when: • Or a Wait message is received.		
8.3.3.6.3.1.7#6	The Policy Engine shall transition to the PE_SRC_Ready state for a Source Port when: • Or the SenderResponseTimer times out.		
8.3.3.6.3.1.7#7	The Policy Engine shall transition to the PE_PRS_SRC_SNK_Transition to off state when: • An Accept message is received.		
8.3.3.6.3.1.8 PE_PRS_SRC_SNK_Reject Swap			
8.3.3.6.3.1.8#1	On entry to the PE_PRS_SRC_SNK_Reject Swap state the Policy Engine shall request the Protocol Layer to send: • A Reject message if the device is unable to perform a Power Role swap at this time. • A Wait message if further evaluation of the Power Role swap request is required		
8.3.3.6.3.1.8#2	The Policy Engine shall transition to the PE_SRC_Ready state when: • The Reject or Wait message has been sent.		
8.3.3.6.3.2 Policy Engine in Sink to Source Power Role swap State Diagram			
8.3.3.6.3.2#1	Dual_Role Ports that combine Sink and Source capabilities shall comprise Sink and Source Policy Engine state machines.		
8.3.3.6.3.2#2	In addition they shall have the capability to do a Power Role Swap from the PE_SNK_Ready state		
8.3.3.6.3.2#3	In addition they...shall return to default operation on a Hard Reset.		
8.3.3.6.3.2#4	Figure 8_54 shows the additional state diagram required to perform a Power Role Swap from Sink to Source roles and the changes that shall be followed for error and Hard Reset handling.		
8.3.3.6.3.2.1 PE_SNK_Ready			
8.3.3.6.3.2.1#1	The Power Role Swap process shall start only from the PE_SNK_Ready state where power is stable.		
8.3.3.6.3.2.1#2	The Policy Engine shall transition to the PE_PRS_SNK_SRC_Evaluate Swap state when: • A PR_Swap message is received.		
8.3.3.6.3.2.1#3	The Policy Engine shall transition to the PE_PRS_SNK_SRC_Send Swap state when: • The Device Policy Manager indicates that a Power Role Swap is required.		
8.3.3.6.3.2.2 PE_PRS_SNK_SRC_Evaluate Swap			
8.3.3.6.3.2.2#1	On entry to the PE_PRS_SNK_SRC_Evaluate Swap state the Policy Engine shall ask the Device Policy Manager whether a Power Role Swap can be made.		
8.3.3.6.3.2.2#2	The Policy Engine shall transition to the PE_PRS_SNK_SRC_Accept Swap state when: • The Device Policy Manager indicates that a Power Role Swap is ok.		
8.3.3.6.3.2.2#3	The Policy Engine shall transition to the PE_PRS_SNK_SRC_Reject Swap state when:		

R2.0 Assertion #	Description	Test	Comments
	• The Device Policy Manager indicates that a Power Role Swap is not ok.		
8.3.3.6.3.2.3 PE_PRS_SNK_SRC_Accept Swap			
8.3.3.6.3.2.3#1	On entry to the PE_PRS_SNK_SRC_Accept Swap state the Policy Engine shall request the Protocol Layer to send an Accept message.		
8.3.3.6.3.2.3#2	The Policy Engine shall transition to the PE_PRS_SNK_SRC_Transition to off state when: • The Accept message has been sent.		
8.3.3.6.3.2.4 PE_PRS_SNK_SRC_Transition to off			
8.3.3.6.3.2.4#1	On entry to the PE_PRS_SNK_SRC_Transition to off state the Policy Engine shall initialize and run the PSSourceOffTimer and then request the Device Policy Manager to turn off the Sink.		
8.3.3.6.3.2.4#2	The Policy Engine shall transition to the PE_SRC_Hard_Reset or PE_SNK_Hard_Reset state depending on its default role as either a Source or Sink Port (see Section 6.7.2) when: · The PSSourceOffTimer times out.		
8.3.3.6.3.2.4#3	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: · The Source Port in a Provider or Provider/Consumer shall go to PE_SRC_Hard_Reset.		
8.3.3.6.3.2.4#4	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: · The Source Port in a Consumer/Provider shall go to PE_SNK_Hard_Reset i.e. revert to default operation as Sink Port.		
8.3.3.6.3.2.4#5	The Policy Engine shall transition to the PE_PRS_SNK_SRC_Source_on state when: · A PS_RDY message is received and · This is not a [USBType_C1.0] DRP		
8.3.3.6.3.2.4#6	The Policy Engine shall transition to the PE_PRS_SNK_SRC_Assert_Rp state when: · A PS_RDY message is received and · This is a [USBType_C1.0] DRP		
8.3.3.6.3.2.5 PE_PRS_SNK_SRC_Assert_Rp			
8.3.3.6.3.1.5#1	On entry to the PE_PRS_SNK_SRC_Assert_Rp state the Policy Engine shall request the Device Policy Manager to change the resistor asserted on the CC wire from Rd to Rp.		
8.3.3.6.3.1.5#2	The Policy Engine shall transition to the PE_PRS_SNK_SRC_Source_on state when: · The Device Policy Manager indicates that Rd is asserted.		
8.3.3.6.3.2.6 PE_PRS_SNK_SRC_Source on			
8.3.3.6.3.2.6#1	On entry to the PE_PRS_SNK_SRC_Source on state the Policy Engine shall request the Device Policy Manager to turn on the Source		
8.3.3.6.3.2.6#2	On entry to the PE_PRS_SNK_SRC_Source on state the Policy Engine...shall initialize and run the SourceActivityTimer		
8.3.3.6.3.2.6#3	On exit from the PE_PRS_SNK_SRC_Source on state (except if the exit is due to a SourceActivityTimer timeout) the Policy Engine shall send a PS_RDY message.		
8.3.3.6.3.2.6#4	The Policy Engine shall transition to the PE_SRC_Startup state when: • The Source Port has been turned on.		
8.3.3.6.3.2.6#5	The Policy Engine shall transition to the PE_SRC_Hard_Reset or PE_SNK_Hard_Reset state depending on its default role as either a Source or Sink Port (see Section 6.7.2) when: · The PS_RDY message is not sent after retries (a GoodCRC message has not been received). A soft reset shall not be		

R2.0 Assertion #	Description	Test	Comments
	initiated in this case.		
8.3.3.6.3.2.6#6	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: · The Source Port in a Provider or Provider/Consumer shall go to PE_SRC_Hard_Reset.		
8.3.3.6.3.2.6#7	The decision as to whether to go to PE_SRC_Hard_Reset or PE_SNK_Hard_Reset shall depend on the type of device: · The Source Port in a Consumer/Provider shall go to PE_SNK_Hard_Reset i.e. revert to default operation as Sink Port.		
8.3.3.6.3.2.7 PE_PRS_SNK_SRC_Send Swap			
8.3.3.6.3.2.7#1	On entry to the PE_PRS_SNK_SRC_Send Swap state the Policy Engine shall request the Protocol Layer to send a Swap message		
8.3.3.6.3.2.7#2	On entry to the PE_PRS_SNK_SRC_Send Swap state the Policy Engine shall...shall initialize and run the SenderResponseTimer.		
8.3.3.6.3.2.7#3	The Policy Engine shall transition to the PE_SNK_Ready state when: • A Reject message is received.		
8.3.3.6.3.2.7#4	The Policy Engine shall transition to the PE_SNK_Ready state when: • Or a Wait message is received.		
8.3.3.6.3.2.7#5	The Policy Engine shall transition to the PE_SNK_Ready state when: • Or the SenderResponseTimer times out.		
8.3.3.6.3.2.7#6	The Policy Engine shall transition to the PE_PRS_SNK_SRC_Transition to off state when: • An Accept message is received.		
8.3.3.6.3.2.8 PE_PRS_SNK_SRC_Reject Swap			
8.3.3.6.3.2.8#1	On entry to the PE_PRS_SNK_SRC_Reject Swap state the Policy Engine shall request the Protocol Layer to send: • A Reject message if the device is unable to perform a Power Role swap at this time. • A Wait message if further evaluation of the Power Role swap request is required. Note: in this case it is expected that one of the Port Partners will send a PR_Swap message at a later time (see Section 6.3.11.2).		
8.3.3.6.3.2.8#2	The Policy Engine shall transition to the PE_SNK_Ready state when: • The Reject or Wait message has been sent.		
8.3.3.6.3.3.1 PE_DR_SRC_Get_Source_Cap state			
8.3.3.6.3.3.1#1	The Policy Engine shall transition to the PE_DR_SRC_Get_Source_Cap state, from the PE_SRC_Ready state, due to a request to get the remote source capabilities from the Device Policy Manager.		
8.3.3.6.3.3.1#2	On entry to the PE_DR_SRC_Get_Source_Cap state the Policy Engine shall send a Get_Source_Cap message and initialize and run the SenderResponseTimer.		
8.3.3.6.3.3.1#3	On exit from the PE_DR_SRC_Get_Source_Cap state the Policy Engine shall inform the Device Policy Manager of the outcome (capabilities or response timeout).		
8.3.3.6.3.3.1#4	The Policy Engine shall transition back to the PE_SRC_Ready state (see Figure 8_39) when: · A Source Capabilities message is received · Or SenderResponseTimer times out · Or a Reject message is received		
8.3.3.6.3.4.1 PE_DR_SRC_Give_Sink_Cap state			
8.3.3.6.3.4.1#1	The Policy Engine shall transition to the		

R2.0 Assertion #	Description	Test	Comments
	PE_DR_SRC_Give_Sink_Cap state, from the PE_SRC_Ready state, when a Get_Sink_Cap message is received.		
8.3.3.6.3.4.1#2	On entry to the PE_DR_SRC_Give_Sink_Cap state the Policy Engine shall request the present capabilities from the Device Policy Manager and then send a Sink Capabilities message based on these capabilities.		
8.3.3.6.3.4.1#3	The Policy Engine shall transition back to the PE_SRC_Ready state (see Figure 8_39) when: · The Sink Capabilities message has been successfully sent.		
8.3.3.6.3.5.1 PE_DR_SNK_Get_Sink_Cap state			
8.3.3.6.3.5.1#1	The Policy Engine shall transition to the PE_DR_SNK_Get_Sink_Cap state, from the PE_SNK_Ready state, due to a request to get the remote source capabilities from the Device Policy Manager.		
8.3.3.6.3.5.1#2	On entry to the PE_DR_SNK_Get_Sink_Cap state the Policy Engine shall send a Get_Sink_Cap message and initialize and run the SenderResponseTimer.		
8.3.3.6.3.5.1#3	On exit from the PE_DR_SNK_Get_Sink_Cap state the Policy Engine shall shall inform the Device Policy Manager of the outcome (capabilities or response timeout).		
8.3.3.6.3.5.1#4	The Policy Engine shall transition back to the PE_SNK_Ready state (see Figure 8_40 and Figure 8_45) when: · A Source Capabilities message is received · Or SenderResponseTimer times out · Or a Reject message is received		
8.3.3.6.3.6.1 PE_DR_SNK_Give_Source_Cap state			
8.3.3.6.3.6.1#1	The Policy Engine shall transition to the PE_DR_SNK_Give_Source_Cap state, from the PE_SNK_Ready state, when a Get_Source_Cap message is received.		
8.3.3.6.3.6.1#2	On entry to the PE_DR_SNK_Give_Source_Cap state the Policy Engine shall request the present capabilities from the Device Policy Manager and then send a Source Capabilities message based on these capabilities.		
8.3.3.6.3.6.1#3	The Policy Engine shall transition back to the PE_SNK_Ready state (see Figure 8_40 and Figure 8_45) when: · The Source Capabilities message has been successfully sent.		
8.3.3.7.1.1 PE_VCS_DFP_Send_Swap state			
8.3.3.7.1.1#1	On entry to the PE_VCS_Send_Swap state the Policy Engine shall send a VCONN_Swap message and start theSenderResponseTimer.		
8.3.3.7.1.1#2	The Policy Engine shall transition to the PE_VCS_Wait_for_UFP_VCONN state when: · An Accept message is received and · DFP current has VCONN turned on.		
8.3.3.7.1.1#3	The Policy Engine shall transition to the PE_VCS_Turn_On_VCONN state when: · An Accept message is received and · DFP current has VCONN turned off.		
8.3.3.7.1.1#4	The Policy Engine shall transition back to either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when: · A Reject message is received or · A Wait message is received or · The SenderResponseTimer times out.		
8.3.3.7.1.2 PE_VCS_Evaluate_Swap			
8.3.3.7.1.2#1	On entry to the PE_VCS_Evaluate_Swap state the Policy Engine shall request the Device Policy Manager for an evaluation of the VCONN Swap request.		
8.3.3.7.1.2#2	The Policy Engine shall transition to the		

R2.0 Assertion #	Description	Test	Comments
	PE_VCS_Accept_Swap state when: · The Device Policy Manager indicates that a VCONN Swap is ok.		
8.3.3.7.1.2#3	The Policy Engine shall transition to the PE_VCS_Reject_Swap state when: · The Device Policy Manager indicates that a VCONN Swap is ok or · The Device Policy Manager indicates that a VCONN Swap cannot be done at this time.		
8.3.3.7.2.2 PE_VCS_Accept_Swap			
8.3.3.7.1.3#1	On entry to the PE_VCS_Accept_Swap state the Policy Engine shall send an Accept message.		
8.3.3.7.1.3#2	The Policy Engine shall transition to the PE_VCS_Wait_for_VCONN state when: · The Accept message has been sent and · The UFP's VCONN is on.		
8.3.3.7.1.3#3	The Policy Engine shall transition to the PE_VCS_Turn_On_VCONN state when: · The Accept message has been sent and · The UFP's VCONN is off.		
8.3.3.7.1.4 PE_VCS_Reject_Swap			
8.3.3.7.1.4#1	On entry to the PE_VCS_Reject_Swap state the Policy Engine shall request the Protocol Layer to send: · A Reject message if the device is unable to perform a VCONN Swap at this time. · A Wait message if further evaluation of the VCONN Swap request is required.		
8.3.3.7.1.4#2	The Policy Engine shall transition back to either the PE_SRC_Ready or PE_SNK_Ready state when: · The Reject or Wait message has been sent.		
8.3.3.7.1.5 PE_VCS_UFP__Wait_for_DFP_VCONN			
8.3.3.7.1.5#1	On entry to the PE_VCS_Wait_for_VCONN state the Policy Engine shall start the VCONNOnTimer.		
8.3.3.7.1.5#2	The Policy Engine shall transition to the PE_VCS_Turn_Off_VCONN state when: · A PS_RDY message is received.		
8.3.3.7.1.5#3	The Policy Engine shall transition to either the PE_SRC_Hard_Reset or PE_SNK_Hard_Reset state when: · The VCONNOnTimer times out.		
8.3.3.7.1.6 PE_VCS_Turn_Off_VCONN			
8.3.3.7.1.6#1	On entry to the PE_VCS_Turn_Off_VCONN state the Policy Engine shall tell the Device Policy Manager to turn off VCONN.		
8.3.3.7.1.6#2	The Policy Engine shall transition back to the either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when: · The UFP's VCONN is off.		
8.3.3.7.1.7 PE_VCS_Turn_On_VCONN			
8.3.3.7.1.7#1	On entry to the PE_VCS_Turn_On_VCONN state the Policy Engine shall tell the Device Policy Manager to turn on VCONN.		
8.3.3.7.1.7#2	The Policy Engine shall transition to the PE_VCS_Send_PS_Rdy state when: · The UFP's VCONN is on.		
8.3.3.7.1.8 PE_VCS_UFP_Send_PS_Rdy			
8.3.3.7.1.8#1	On entry to the PE_VCS_Send_PS_Rdy the Policy Engine shall send a PS_RDY message.		
8.3.3.7.1.8#2	The Policy Engine shall transition back to the either the PE_SRC_Ready or PE_SNK_Ready state for a UFP when: · The PS_RDY message has been sent.		

R2.0 Assertion #	Description	Test	Comments
8.3.3.8.1.1 PE_UFP_VDM_Get_Identity state			
8.3.3.8.1.1#1	On entry to the PE_UFP_VDM_Get_Identity state the UFP shall request identity information from the Device Policy Manager.		
8.3.3.8.1.1#2	The Policy Engine shall transition to the PE_UFP_VDM_Send_Identity state when: · Identity information is received from the Device Policy Manager.		
8.3.3.8.1.1#3	The Policy Engine shall transition to the PE_UFP_VDM_Get_Identity_NAK state when: · The Device Policy Manager indicates that the response to the Discover Identity request is NAK or BUSY.		
8.3.3.8.1.2 PE_UFP_VDM_Send_Identity state			
8.3.3.8.1.2#1	On entry to the PE_UFP_VDM_Send_Identity state the UFP shall send the Structured VDM Discover Identity ACK Command response.	BMC_PHY_MSG_6 BMC_PHY_MSG_12	
8.3.3.8.1.2#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a UFP when: · The Structured VDM Discover Identity ACK Command response has been sent.	BMC_PHY_MSG_6 BMC_PHY_MSG_12	
8.3.3.8.1.3 PE_UFP_VDM_Get_Identity_NAK state			
8.3.3.8.1.3#1	"On entry to the PE_UFP_VDM_Get_Identity_NAK state the Policy Engine shall send a Structured VDM Discover Identity NAK or BUSY Command response as indicated by the Device Policy Manager.		
8.3.3.8.1.3#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a UFP when: · The Structured VDM Discover Identity NAK or BUSY Command response has been sent.		
8.3.3.8.2.1 PE_UFP_VDM_Get_SVIDs state			
8.3.3.8.2.1#1	On entry to the PE_UFP_VDM_Get_SVIDs state the UFP shall request SVIDs information from the Device Policy Manager.		
8.3.3.8.2.1#2	The Policy Engine shall transition to the PE_UFP_VDM_Send_SVIDs state when: · SVIDs information is received from the Device Policy Manager.		
8.3.3.8.2.1#3	The Policy Engine shall transition to the PE_UFP_VDM_Get_SVIDs_NAK state when: · The Device Policy Manager indicates that the response to the Discover Identity request is NAK or BUSY.		
8.3.3.8.2.2 PE_UFP_VDM_Send_SVIDs state			
8.3.3.8.2.2#1	On entry to the PE_UFP_VDM_Send_SVIDs state the UFP shall send the Structured VDM Discover SVIDs ACK Command response.		
8.3.3.8.2.2#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a UFP when: · The Structured VDM Discover SVIDs ACK Command response has been sent.		
8.3.3.8.2.3 PE_UFP_VDM_Get_SVIDs_NAK state			
8.3.3.8.2.3#1	On entry to the PE_UFP_VDM_Get_SVIDs_NAK state the Policy Engine shall send a Structured VDM Discover SVIDs NAK or BUSY Command response as indicated by the Device Policy Manager.		
8.3.3.8.2.3#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a UFP when: · The Structured VDM Discover SVIDs NAK or BUSY Command response has been sent.		
8.3.3.8.3.1 PE_UFP_VDM_Get_Modes state			
8.3.3.8.3.1#1	On entry to the PE_UFP_VDM_Get_Modes state the UFP		

R2.0 Assertion #	Description	Test	Comments
	shall request Modes information from the Device Policy Manager.		
8.3.3.8.3.1#2	The Policy Engine shall transition to the PE_UFP_VDM_Send_Modes state when: · Modes information is received from the Device Policy Manager.		
8.3.3.8.3.1#3	The Policy Engine shall transition to the PE_UFP_VDM_Get_Modes_NAK state when: · The Device Policy Manager indicates that the response to the Discover Identity request is NAK or BUSY.		
8.3.3.8.3.2 PE_UFP_VDM_Send_Modes state			
8.3.3.8.3.2#1	On entry to the PE_UFP_VDM_Send_Modes state the UFP shall send the Structured VDM Discover Modes ACK Command response.		
8.3.3.8.3.2#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a UFP when: · The Structured VDM Discover Modes ACK Command response has been sent.		
8.3.3.8.3.3 PE_UFP_VDM_Send_Modes state			
8.3.3.8.3.3#1	On entry to the PE_CBL_Get_Modes_NAK state the Policy Engine shall send a Structured VDM Discover Modes NAK or BUSY Command response as indicated by the Device Policy Manager.		
8.3.3.8.3.3#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a UFP when: · The Structured VDM Discover Modes NAK or BUSY Command response has been sent.		
8.3.3.8.4.1 PE_UFP_VDM_Evaluate_Mode_Entry state			
8.3.3.8.4.1#1	On Entry to the PE_UFP_VDM_Evaluate_Mode_Entry state the Policy Engine shall request the Device Policy Manager to evaluate the Enter Mode Command request and enter the Mode indicated in the Command request if the request is acceptable.		
8.3.3.8.4.1#2	The Policy Engine shall transition to the PE_UFP_VDM_Mode_Entry_ACK state when: · The Device Policy Manager indicates that the Mode has been entered.		
8.3.3.8.4.1#3	The Policy Engine shall transition to the PE_UFP_VDM_Mode_Entry_NAK state when: · The Device Policy Manager indicates that the Command response to the Enter Mode Command request is NAK.		
8.3.3.8.4.2 PE_UFP_VDM_Mode_Entry_ACK state			
8.3.3.8.4.2#1	On entry to the PE_UFP_VDM_Mode_Entry_ACK state the Policy Engine shall send a Structured VDM Enter Mode ACK Command response.		
8.3.3.8.4.2#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a UFP when: · The Structured VDM Enter Mode ACK Command response has been sent.		
8.3.3.8.4.3 PE_UFP_VDM_Mode_Entry_NAK state			
8.3.3.8.4.3#1	On entry to the PE_UFP_VDM_Mode_Entry_NAK state the Policy Engine shall send a Structured VDM Enter Mode NAK Command response as indicated by the Device Policy Manager.		
8.3.3.8.4.3#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a UFP when: · The Structured VDM Enter Mode NAK Command response has been sent.		
8.3.3.8.5.1 PE_UFP_VDM_Mode_Exit state			

R2.0 Assertion #	Description	Test	Comments
8.3.3.8.5.1#1	On entry to the PE_UFP_VDM_Mode_Exit state the Policy Engine shall request the Device Policy Manager to exit the Mode indicated in the Command.		
8.3.3.8.5.1#2	The Policy Engine shall transition to the PE_UFP_VDM_Mode_Exit_ACK state when: | The Device Policy Manger indicates that the Mode has been exited.		
8.3.3.8.5.1#3	The Policy Engine shall transition to the PE_UFP_VDM_Mode_Exit_NAK state when: · The Device Policy Manager indicates that the Command response to the Exit Mode Command request is NAK.		
8.3.3.8.5.1 PE_UFP_VDM_Mode_Exit state			
8.3.3.8.5.2#1	On entry to the PE_UFP_VDM_Mode_Exit_ACK state the Policy Engine shall send a Structured VDM Exit Mode ACK Command response.		
8.3.3.8.5.2#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a UFP when: · The Structured VDM Exit Mode ACK Command response has been sent.		
8.3.3.8.5.3 PE_UFP_VDM_Mode_Exit_ACK state			
8.3.3.8.5.3#1	On entry to the PE_UFP_VDM_Mode_Exit_ACK state the Policy Engine shall send a Structured VDM Exit Mode ACK Command response.		
8.3.3.8.5.3#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a UFP when: · The Structured VDM Exit Mode ACK Command response has been sent.		
8.3.3.8.6.1 PE_UFP_VDM_Attention_Request			
8.3.3.8.6.1#1	On entry to the PE_UFP_VDM_Attention_Request state the Policy Engine shall send an Attention Command request.		
8.3.3.8.6.1#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when: · The Attention Command request has been sent.		
8.3.3.9.1.1 PE_DFP_VDM_Identity_Request state			
8.3.3.9.1.1#1	On entry to the PE_DFP_VDM_Identity_Request state the Policy Engine shall send a Structured VDM Discover Identity Command request and shall start the VDMResponseTimer.		
8.3.3.9.1.1#2	The Policy Engine shall transition to the PE_DFP_VDM_Identity_ACKed state when: · A Structured VDM Discover Identity ACK Command response is received.	BMC_PHY_MSG_6 BMC_PHY_MSG_12	
8.3.3.9.1.1#3	The Policy Engine shall transition to the PE_DFP_UFP_VDM_Identity_NAKed state when: · A Structured VDM Discover Identity NAK or BUSY Command response is received or · The VDMResponseTimer times out.		
8.3.3.9.1.2 PE_DFP_VDM_Identity_ACKed state			
8.3.3.9.1.2#1	On entry to the PE_DFP_VDM_Identity_ACKed state the Policy Engine shall inform the Device Policy Manager of the Identity information.		
8.3.3.9.1.2#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when: · The Device Policy Manager has been informed.		
8.3.3.9.1.3 PE_DFP_VDM_Identity NAKed state			
8.3.3.9.1.3#1	On entry to the PE_DFP_VDM_Identity_NAKed state the Policy Engine shall inform the Device Policy Manager of the result (NAK, BUSY or timeout).		
8.3.3.9.1.3#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when:		

R2.0 Assertion #	Description	Test	Comments
	· The Device Policy Manager has been informed.		
8.3.3.9.2.1 PE_DFP_VDM_SVIDs_Request state			
8.3.3.9.3.1#1	On entry to the PE_DFP_VDM_SVIDs_Request state the Policy Engine shall send a Structured VDM Discover SVIDs Command request and shall start the VDMResponseTimer.		
8.3.3.9.3.1#2	The Policy Engine shall transition to the PE_DFP_VDM_SVIDs_ACKed state when: · A Structured VDM Discover SVIDs ACK Command response is received.		
8.3.3.9.3.1#3	The Policy Engine shall transition to the PE_DFP_VDM_SVIDs_NAKed state when: · A Structured VDM Discover SVIDs NAK or BUSY Command response is received or · The VDMResponseTimer times out.		
8.3.3.9.3.2 PE_DFP_VDM_SVIDs_ACKed state			
8.3.3.9.3.2#1	On entry to the PE_DFP_VDM_SVIDs_ACKed state the Policy Engine shall inform the Device Policy Manager of the SVIDs information.		
8.3.3.9.3.2#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when: · The Device Policy Manager has been informed.		
8.3.3.9.3.3 PE_DFP_VDM_SVIDs_NAKed state			
8.3.3.9.3.3#1	On entry to the PE_DFP_VDM_SVIDs_NAKed state the Policy Engine shall inform the Device Policy Manager of the result (NAK, BUSY or timeout).		
8.3.3.9.3.3#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when: · The Device Policy Manager has been informed.		
8.3.3.9.4.1 PE_DFP_VDM_Modes Request			
8.3.3.9.4.1#1	On entry to the PE_DFP_VDM_Modes_Request state the Policy Engine shall send a Structured VDM Discover Modes Command request and shall start the VDMResponseTimer.		
8.3.3.9.4.1#2	The Policy Engine shall transition to the PE_DFP_VDM_Modes_ACKed state when: · A Structured VDM Discover Modes ACK Command response is received.		
8.3.3.9.4.1#3	The Policy Engine shall transition to the PE_DFP_VDM_Modes_NAKed state when: · A Structured VDM Discover Modes NAK or BUSY Command response is received or · The VDMResponseTimer times out.		
8.3.3.9.4.2 DFP Modes ACKed			
8.3.3.9.4.2#1	On entry to the PE_DFP_VDM_Modes_ACKed state the Policy Engine shall inform the Device Policy Manager of the Modes information.		
8.3.3.9.4.2#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when: · The Device Policy Manager has been informed.		
8.3.3.9.4.3 DFP Modes NAKed			
8.3.3.9.4.3#1	On entry to the PE_DFP_VDM_Modes_NAKed state the Policy Engine shall inform the Device Policy Manager of the result (NAK, BUSY or timeout).		
8.3.3.9.4.3#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when: · The Device Policy Manager has been informed.		
8.3.3.9.5.1 PE_DFP_VDM_Mode_Entry_Request state			
8.3.3.9.5.1#1	On entry to the PE_DFP_VDM_Mode_Entry_Request state the Policy Engine shall send 1 a Structured VDM Enter Mode Command request and shall start the VDMModeEntryTimer.		

R2.0 Assertion #	**Description**	**Test**	**Comments**
8.3.3.9.5.1#2	The Policy Engine shall transition to the PE_DFP_VDM_Mode_Entry_ACKed state when: · A Structured VDM Enter Mode ACK Command response is received.		
8.3.3.9.5.1#3	The Policy Engine shall transition to the PE_DFP_VDM_Mode_Entry_NAKed state when: · A Structured VDM Enter Mode NAK or BUSY Command response is received or · The VDMModeEntryTimer times out.		
8.3.3.9.5.2 PE_DFP_VDM_Mode_Entry_ACKed state			
8.3.3.9.5.2#1	On entry to the PE_DFP_VDM_Mode_Entry_ACKed state the Policy Engine shall request the Device Policy Manager to enter the Mode.		
8.3.3.9.5.2#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when: · The Mode has been entered.		
8.3.3.9.5.3 PE_DFP_VDM_Mode_Entry_NAKed state			
8.3.3.9.5.3#1	On entry to the PE_DFP_VDM_Mode_Entry_NAKed state the Policy Engine shall inform the Device Policy Manager of the result (NAK, BUSY or timeout).		
8.3.3.9.5.3#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when: · The Device Policy Manager has been informed.		
8.3.3.9.6.1 PE_DFP_VDM_Mode_Exit_Request state			
8.3.3.9.6.1#1	On entry to the PE_DFP_VDM_Mode_Exit_Request state the Policy Engine shall send a Structured VDM Exit Mode Command request and shall start the VDMModeExitTimer.		
8.3.3.9.6.1#2	The Policy Engine shall transition to the PE_DFP_VDM_Mode_Exit_ACKed state when: · A Structured VDM Exit Mode ACK Command response is received.		
8.3.3.9.6.1#3	The Policy Engine shall transition to either the PE_SRC_Hard_Reset or PE_PC_SNK_Hard_Reset state for a DFP when: · A Structured VDM Exit Mode NAK or BUSY Command response is received or · The VDMModeExitTimer times out.		
8.3.3.9.6.2 PE_DFP_VDM_DFP_Mode_Exit_ACKed state			
8.3.3.9.6.2#1	On Exit to the PE_DFP_VDM_Mode_Exit_ACKed state the Policy Engine shall request the Device Policy Manager to exit the Mode.		
8.3.3.9.6.2#2	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state for a DFP when: · The Device Policy Manager has been informed.		
8.3.3.9.7.1 PE_DFP_VDM_Attention_Request			
8.3.3.9.7.1#1	On entry to the PE_DFP_VDM_Attention_Request state the Policy Engine shall inform the Device Policy Manager of the attention request.		
8.3.3.9.7.1#2	The Policy Engine shall transition to either the PE_SRC_Ready		

R2.0 Assertion #	Description	Test	Comments
	or PE_SNK_Ready state for a DFP when: · The Device Policy Manager has been informed.		
8.3.3.10.2.1 PE_CBL_Get_Identity state			
8.3.3.10.2.1#1	On entry to the PE_CBL_Get_Identity state the Cable shall request identity information from the Device Policy Manager.		
8.3.3.10.2.1#2	The Policy Engine shall transition to the PE_CBL_Send_Identity state when: · Identity information is received from the Device Policy Manager.		
8.3.3.10.2.2 PE_CBL_Send_Identity state			
8.3.3.10.2.2#1	On entry to the PE_CBL_Send_Identity state the Cable shall send the Structured VDM Discover Identity ACK Command response.	BMC_PHY_MSG_6 BMC_PHY_MSG_12	
8.3.3.10.2.2#2	The Policy Engine shall transition to the PE_CBL_Ready state when: · The Structured VDM Discover Identity ACK Command response has been sent.	BMC_PHY_MSG_6 BMC_PHY_MSG_12	
8.3.3.10.2.3 PE_CBL_Get_Identity NAK			
8.3.3.10.2.3#1	On entry to the PE_CBL_Get_Identity_NAK state the Policy Engine shall send a Structured VDM Discover Identity NAK or BUSY Command response as indicated by the Device Policy Manager.	BMC_PHY_MSG_6 BMC_PHY_MSG_12	
8.3.3.10.2.3#2	The Policy Engine shall transition to the PE_CBL_Ready state when: · The Structured VDM Discover Identity NAK or BUSY Command response has been sent.	BMC_PHY_MSG_6 BMC_PHY_MSG_12	
8.3.3.10.3.1 PE_CBL_Get_SVIDs state			
8.3.3.10.3.1#1	On entry to the PE_CBL_Get_SVIDs state the Cable shall request SVIDs information from the Device Policy Manager.		
8.3.3.10.3.1#2	The Policy Engine shall transition to the PE_CBL_Send_SVIDs state when: · SVIDs information is received from the Device Policy Manager.		
8.3.3.10.3.2 PE_CBL_Send_SVIDs state			
8.3.3.10.3.2#1	On entry to the PE_CBL_Send_SVIDs state the Cable shall send the Structured VDM Discover SVIDs ACK Command response.		
8.3.3.10.3.2#2	The Policy Engine shall transition to the PE_CBL_Ready state when: · The Structured VDM Discover SVIDs ACK Command response has been sent.		
8.3.3.10.4.1 PE_CBL_Get_Modes state			
8.3.3.10.4.1#1	On entry to the PE_CBL_Get_Modes state the Cable shall request Modes information from the Device Policy Manager.		
8.3.3.10.4.1#2	The Policy Engine shall transition to the PE_CBL_Send_Modes state when: · Modes information is received from the Device Policy Manager.		
8.3.3.10.4.2 PE_CBL_Send_Modes state			
8.3.3.10.4.2#1	On entry to the PE_CBL_Send_Modes state the Cable shall send the Structured VDM Discover Modes ACK Command response.		
8.3.3.10.4.2#2	The Policy Engine shall transition to the PE_CBL_Ready state when: · The Structured VDM Discover Modes ACK Command response has been sent.		
8.3.3.10.4.3 PE_CBL_Get_Modes NAK			
8.3.3.10.4.3#1	On entry to the PE_CBL_Get_Modes_NAK state the Policy		

R2.0 Assertion #	Description	Test	Comments
	Engine shall send a Structured VDM Discover Modes NAK or BUSY Command response as indicated by the Device Policy Manager.		
8.3.3.10.4.3#2	The Policy Engine shall transition to the PE_CBL_Ready state when: · The Structured VDM Discover Modes NAK or BUSY Command response has been sent.		
8.3.3.10.5.1 PE_CBL_Evaluate_Mode_Entry state			
8.3.3.10.5.1#1	On Entry to the PE_CBL_Evaluate_Mode_Entry state the Policy Engine shall request the Device Policy Manager to evaluate the Enter Mode Command request and enter the Mode indicated in the Command request if the request is acceptable.		
8.3.3.10.5.1#2	The Policy Engine shall transition to the PE_CBL_Mode_Entry_ACK state when: · The Device Policy Manager indicates that the Mode has been entered.		
8.3.3.10.5.1#3	The Policy Engine shall transition to the PE_CBL_Mode_Entry_NAK state when: · The Device Policy Manager indicates that the response to the Mode request is NAK.		
8.3.3.10.5.2 PE_CBL_Mode_Entry_ACK state			
8.3.3.10.5.2#1	On entry to the PE_CBL_Mode_Entry_ACK state the Policy Engine shall send a Structured VDM Enter Mode ACK Command response.		
8.3.3.10.5.2#2	The Policy Engine shall transition to the PE_CBL_Ready state when: · The Structured VDM Enter Mode ACK Command response has been sent.		
8.3.3.10.5.3 PE_CBL_Mode_Entry_NAK state			
8.3.3.10.5.3#1	On entry to the PE_CBL_Mode_Entry_NAK state the Policy Engine shall send a Structured VDM Enter Mode NAK Command response as indicated by the Device Policy Manager.		
8.3.3.10.5.3#2	The Policy Engine shall transition to the PE_CBL_Ready state when: · The Structured VDM Enter Mode NAK Command response has been sent.		
8.3.3.10.6.1 PE_CBL_Mode_Exit state			
8.3.3.10.6.1#1	On entry to the PE_CBL_Mode_Exit state the Policy Engine shall request the Device Policy Manager to exit the Mode indicated in the Command.		
8.3.3.10.6.1#2	The Policy Engine shall transition to the PE_CBL_Mode_Exit_ACK state when: · The Device Policy Manager indicates that the Mode has been exited.		
8.3.3.10.6.1#3	The Policy Engine shall transition to the PE_CBL_Mode_Exit_NAK state when: · The Device Policy Manager indicates that the Command response to the Exit Mode Command request is NAK.		
8.3.3.10.6.2 PE_CBL_Mode_Exit_ACK state			
8.3.3.10.6.2#1	On entry to the PE_CBL_Mode_Exit_ACK state the Policy Engine shall send a Structured VDM Exit Mode ACK Command response.		
8.3.3.10.6.2#2	The Policy Engine shall transition to the PE_CBL_Ready state when: · The Structured VDM Exit Mode ACK Command response has been sent.		
8.3.3.10.6.3 PE_UFP_VDM_Mode_Exit_NAK state			

R2.0 Assertion #	Description	Test	Comments
8.3.3.10.6.3#1	On entry to the PE_CBL_Mode_Exit_NAK state the Policy Engine shall send a Structured VDM Exit Mode NAK Command response as indicated by the Device Policy Manager.		
8.3.3.10.6.3#2	The Policy Engine shall transition to the PE_CBL_Ready state when: · The Structured VDM Exit Mode NAK Command response has been sent.		
8.3.3.10.7.1 PE_CBL_Soft_Reset state			
8.3.3.10.7.1#1	The PE_CBL_Soft_Reset state shall be entered from any state when a Reset message is received from the Protocol Layer.		
8.3.3.10.7.1#2	On entry to the PE_CBL_Soft_Reset state the Policy Engine shall reset the Protocol Layer in the Cable Plug and shall then request the Protocol Layer to send an Accept message.		
8.3.3.10.7.1#3	The Policy Engine shall transition to the PE_CBL_Ready state when: · The Accept message has been sent.		
8.3.3.10.7.1#4	The Policy Engine shall transition to the PE_CBL_Hard_Reset state when: · The Protocol Layer indicates that a transmission error has occured.		
8.3.3.10.8.1 PE_CBL_Hard_Reset state			
8.3.3.10.8.1#1	The PE_CBL_Hard_Reset state shall be entered from any state when either Hard Reset signaling or Cable Reset signaling is detected.		
8.3.3.10.8.1#2	On entry to the PE_CBL_Hard_Reset state the Policy Engine shall reset the Cable Plug (equivalent to a power cycle).		
8.3.3.10.8.1#3	The Policy Engine shall transition to the PE_CBL_Ready state when: · The Cable Plug reset is complete.		
8.3.3.10.9.1 PE_DFP_CBL_Send_Soft_Reset state			
8.3.3.10.9.1#1	The PE_DFP_CBL_Send_Soft_Reset state shall be entered from any state when a Protocol Error is detected by the Protocol Layer (see Section 6.7.1) when a Message has not been sent after retries while communicating with a Cable Plug or whenever the Device Policy Manager directs a Soft Reset .		
8.3.3.10.9.1#2	On entry to the PE_DFP_CBL_Send_Soft_Reset state the Policy Engine shall request the Protocol Layer to perform a Soft Reset, then shall send a Soft Reset message to the Cable Plug, and initialize and run the SenderResponseTimer.		
8.3.3.10.9.1#3	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state, depending on the DFP's Power Role, when: · An Accept message has been received.		
8.3.3.10.9.1#4	The Policy Engine shall transition to the PE_DFP_CBL_Send_Cable_Reset state when: · A SenderResponseTimer timeout occurs · Or the Protocol Layer indicates that a transmission error has occurred.		
8.3.3.10.9.2 PE_DFP_CBL_Send_Cable_Reset state			
8.3.3.10.9.2#1	The PE_DFP_CBL_Send_Cable_Reset state shall be entered from any state when the Device Policy Manager requests a Cable Reset.		
8.3.3.10.9.2#2	On entry to the PE_DFP_CBL_Send_Cable_Reset state the Policy Engine shall request the Protocol Layer to send Cable Reset signaling.		
8.3.3.10.9.2#3	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state, depending on the DFP's Power Role,		

R2.0 Assertion #	Description	Test	Comments
	when: · Cable Reset signaling has been sent.		
8.3.3.10.10.1 PE_UFP_CBL_Send_Soft_Reset state			
8.3.3.10.10.1#1	The PE_UFP_CBL_Send_Soft_Reset state shall be entered from any state when a Protocol Error is detected by the Protocol Layer, when a Message has not been sent after retries while communicating with a Cable Plug or whenever the Device Policy Manager directs a Soft Reset.		
8.3.3.10.10.1#2	On entry to the PE_UFP_CBL_Send_Soft_Reset state the Policy Engine shall request the Protocol Layer to perform a Soft Reset, then shall send a Soft Reset message to the Cable Plug, and initialize and run the SenderResponseTimer.		
8.3.3.10.10.1#3	The Policy Engine shall transition to either the PE_SRC_Ready or PE_SNK_Ready state, depending on the DFP's Power Role, when: · An Accept message has been received · Or a SenderResponseTimer timeout occurs · Or the Protocol Layer indicates that a transmission error has occurred		
8.3.3.10.11.1 PE_SRC_VDM_Identity_Request state			
8.3.3.10.11.1#1	The Policy Engine shall transition to the PE_SRC_VDM_Identity_Request state from the PE_SRC_Startup state when: · The Device Policy Manager requests the discovery of the identity of the Port Partner or a Cable Plug.		
8.3.3.10.11.1#2	Even though there has been a transition out of the PE_SRC_Discovery state the SourceCapabilityTimer shall continue to run during the states shown in Figure 8-81 and shall not be initialized on re-entry to PE_SRC_Discovery.		
8.3.3.10.11.1#3	The Policy Engine shall transition to the PE_SRC_VDM_Identity_Request state from the PE_SRC_Discovery state when: · The Device Policy Manager requests the discovery of the identity of the Port Partner or a Cable Plug and · The DiscoverIdentityCounter < nDiscoverIdentityCount.		
8.3.3.10.11.1#4	On entry to the PE_SRC_VDM_Identity_Request state the Policy Engine shall send a Structured VDM Discover Identity Command request and shall start the VDMResponseTimer.		
8.3.3.10.11.1#5	The Policy Engine shall transition to the PE_SRC_VDM_Identity_ACKed state when: · A Structured VDM Discover Identity ACK Command response is received.	BMC_PHY_MSG_6 BMC_PHY_MSG_12	
8.3.3.10.11.1#6	The Policy Engine shall transition to the PE_SRC_VDM_Identity_NAKed state when: · A Structured VDM Discover Identity NAK or BUSY Command response is received or · The VDMResponseTimer times out.		
8.3.3.10.11.2 PE_SRC_VDM_Identity_ACKed state			
8.3.3.10.11.2#1	On entry to the PE_SRC_VDM_Identity_ACKed state the Policy Engine shall inform the Device Policy Manager of the Identity information.		
8.3.3.10.11.2#2	The Policy Engine shall transition to the PE_SRC_Ready state when: · The Device Policy Manager has been informed.		
8.3.3.10.11.3 PE_SRC_VDM_Identity_NAKed state			
8.3.3.10.11.3#1	On entry to the PE_SRC_VDM_Identity_NAKed state the Policy Engine shall inform the Device Policy Manager of the result (NAK, BUSY or timeout).		
8.3.3.10.11.3#2	The Policy Engine shall transition to the PE_SRC_Ready state		

R2.0 Assertion #	Description	Test	Comments
	when: · The Device Policy Manager has been informed.		
8.3.3.11.1 BIST Receive Mode State Diagram			
8.3.3.11.1#1	Transitions to PE_BIST_Receive_Mode state shall be from either the PE_SRC_Ready or PE_SNK_Ready state or PE_CBL_Ready state.		
8.3.3.11.1.1 PE_BIST_Receive_Mode state			
8.3.3.11.1.1#1	The Source or Sink shall enter the PE_BIST_Receive_Mode state from either the PE_SRC_Ready or PE_SNK_Ready or PE_CBL_Ready state when: · A BIST message is received with a BIST Receiver Mode BIST data object and VBUS is at vSafe5V.		
8.3.3.11.1.1#2	On entry to the PE_BIST_Receive_Mode state the Policy Engine shall tell the Protocol Layer to go to BIST Receive Mode.		
8.3.3.11.1.1#3	The Policy Engine shall transition to the PE_BIST_Frame_Received state when: · A Test Frame is received.		
8.3.3.11.1.1#4	The Policy Engine shall transition to either the PE_SRC_Transition_to_default state or PE_SNK_Transition_to_default state or PE_CBL_State (as appropriate) when: · Hard Reset signaling is received.		
8.3.3.11.1.2 PE_BIST_Frame_Received state			
8.3.3.11.1.2#1	On entry to the PE_BIST_Frame_Received state the Policy Engine shall consume a BIST Transmit Test Frame if one has been received.		
8.3.3.11.1.2#2	The Policy Engine shall transition back to the PE_BIST_Frame_Received state when: · The BIST Receive Test Frame has been received.		
8.3.3.11.1.2#3	The Policy Engine shall transition to either the PE_SRC_Transition_to_default state or PE_SNK_Transition_to_default state ro PE_CBL_Ready state (as appropriate) when: · Hard Reset signaling is received.		
8.3.3.11.2 BIST Transmit Mode State Diagram			
8.3.3.11.2#1	Transitions to PE_BIST_Transmit_Mode state shall be from either the PE_SRC_Ready or PE_SNK_Ready or PE_CBL_Ready state.		
8.3.3.11.2.1 PE_BIST_Transmit_Mode state			
8.3.3.11.2.1#1	The Source or Sink shall enter the PE_BIST_Transmit_Mode state from either the PE_SRC_Ready, PE_SNK_Ready state or PE_CBL_Ready state when: · A BIST message is received with a BIST Transmit Mode BIST data object and VBUS is at vSafe5V.		
8.3.3.11.2.1#2	On entry to the PE_BIST_Transmit_Mode state the Policy Engine shall tell the Protocol Layer to go to BIST Transmit Mode.		
8.3.3.11.2.1#3	The Policy Engine shall transition to the PE_BIST_Send_Frame state when: · The Protocol Layer is in BIST Transmit Mode.		
8.3.3.11.2.1#4	The Policy Engine shall transition to either the PE_SRC_Transition_to_default state, PE_SNK_Transition_to_default state or PE_CBL_Ready state (as appropriate) when: · Hard Reset signaling is received.		
8.3.3.11.2.2 PE_BIST_Send_Frame state			
8.3.3.11.2.2#1	On entry to the PE_BIST_Send_Frame state the Policy Engine shall tell the Protocol Layer to send the next BIST Transmit		

R2.0 Assertion #	Description	Test	Comments
	Test Frame in the PRBS sequence.		
8.3.3.11.2.2#2	The Policy Engine shall transition back to the PE_BIST_Send_Frame state when: · The BIST Transmit Test Frame has been sent (a BIST message with a BISTErrorCounter data object has been received).		
8.3.3.11.2.2#3	The Policy Engine shall transition to either the PE_SRC_Transition_to_default state or PE_SNK_Transition_to_default state (as appropriate) when: · Hard Reset signaling is received.		
8.3.3.11.3 BIST Carrier Mode and Eye Pattern State Diagram			
8.3.3.11.3#1	Transitions to any of the Test Mode states shall be from either the PE_SRC_Ready or PE_SNK_Ready state.		
8.3.3.11.3.1 BIST Eye Pattern			
8.3.3.11.3.1#1	The Source, Sink or Cable Plug shall enter the PE_BIST_Eye_Pattern_Mode state from either the PE_SRC_Ready, PE_SNK_Ready state or PE_CBL_Ready state when: · A BIST message is received with a BIST Eye Pattern data object and · VBUS is at vSafe5V.		
8.3.3.11.3.1#2	On entry to the PE_BIST_Eye_Pattern_Mode state the Policy Engine shall tell the Protocol Layer to go to BIST Eye Pattern Test Mode and shall initialize and run the BISTContModeTimer.		
8.3.3.11.3.1#3	The Policy Engine shall transition to either the PE_SRC_Transition_to_default state, PE_SNK_Transition_to_default state or PE_CBL_Ready state (as appropriate) when: · The BISTContModeTimer times out.		
8.3.3.11.3.2 BIST Carrier Mode 0			
8.3.3.11.3.2#1	The Source, Sink or Cable Plug shall enter the PE_BIST_Carrier_Mode_0 state from either the PE_SRC_Ready, PE_SNK_Ready state or PE_CBL_Ready state when: · A BIST message is received with a BIST Carrier Mode 0 data object and · VBUS is at vSafe5V.		
8.3.3.11.3.2#2	On entry to the PE_BIST_Carrier_Mode_0 state the Policy Engine shall tell the Protocol Layer to go to BIST Carrier Purity Mode 0 and shall initialize and run the BISTContModeTimer.		
8.3.3.11.3.2#3	The Policy Engine shall transition to either the PE_SRC_Transition_to_default state, PE_SNK_Transition_to_default state or PE_CBL_Ready state (as appropriate) when: · The BISTContModeTimer times out.		
8.3.3.11.3.3 BIST Carrier Mode 1			
8.3.3.11.3.3#1	The Source, Sink or Cable Plug shall enter the PE_BIST_Carrier_Mode_1 state from either the PE_SRC_Ready, PE_SNK_Ready state or PE_CBL_Ready state when: · A BIST message is received with a BIST Carrier Mode 1 data object and · VBUS is at vSafe5V.		
8.3.3.11.3.3#2	On entry to the PE_BIST_Carrier_Mode_1 state the Policy Engine shall tell the Protocol Layer to go to BIST Carrier Purity Mode 1 and shall initialize and run the BISTContModeTimer.		
8.3.3.11.3.3#3	The Policy Engine shall transition to either the		

R2.0 Assertion #	Description	Test	Comments
	PE_SRC_Transition_to_default state, PE_SNK_Transition_to_default state or PE_CBL_Ready state (as appropriate) when: · The BISTContModeTimer times out.		
8.3.3.11.3.4 BIST Carrier Mode 2			
8.3.3.11.3.4#1	The Source, Sink or Cable Plug shall enter the PE_BIST_Carrier_Mode_2 state from either the PE_SRC_Ready, PE_SNK_Ready state or PE_CBL_Ready state when: · A BIST message is received with a BIST Carrier Mode 2 data object and · VBUS is at vSafe5V.		
8.3.3.11.3.4#2	On entry to the PE_BIST_Carrier_Mode_2 state the Policy Engine shall tell the Protocol Layer to go to BIST Carrier Purity Mode 2 and shall initialize and run the BISTContModeTimer.		
8.3.3.11.3.4#3	The Policy Engine shall transition to either the PE_SRC_Transition_to_default state, PE_SNK_Transition_to_default state or PE_CBL_Ready state (as appropriate) when: · The BISTContModeTimer times out.		
8.3.3.11.3.5 BIST Carrier Mode 3			
8.3.3.11.3.5#1	The Source, Sink or Cable Plug shall enter the PE_BIST_Carrier_Mode_3 state from either the PE_SRC_Ready, PE_SNK_Ready state or PE_CBL_Ready state when: · A BIST message is received with a BIST Carrier Mode 3 data object and · VBUS is at vSafe5V.		
8.3.3.11.3.5#2	On entry to the PE_BIST_Carrier_Mode_3 state the Policy Engine shall tell the Protocol Layer to go to BIST Carrier Mode 3 and shall initialize and run the BISTContModeTimer.		
8.3.3.11.3.5#3	The Policy Engine shall transition to either the PE_SRC_Transition_to_default state, PE_SNK_Transition_to_default state or PE_CBL_Ready state (as appropriate) when: · The BISTContModeTimer times out.		
8.3.3.12.1 ErrorRecovery State			
8.3.3.12.1#1	The ErrorRecovery state shall be entered when there are errors on Type_C Ports which cannot be recovered by Hard Reset.		
8.3.3.12.1#2	The ErrorRecovery state shall map to Type-C ErrorRecovery state operation as defined in the [USBType-C 1.0] specification, including any other state transitions mandated in cases where Type-C ErrorRecovery is not supported.		
8.3.3.12.1#3	On entry to the ErrorRecovery state the Contract and PD Connection shall be ended.		

7.7.2 Recommended (Should)

R2.0 Assertion #	Description	Test	Comments
Subsection 8.2 Device Policy Manager			
8.2#r1	Within multiport devices it is also permitted to have a combination of USB Power Delivery and non_USB Power Delivery ports which should all be managed by the Device Policy Manager.		
8.2.5.1 Managing the Power Reserve			
8.2.5.1#r1	For example, a mobile device with a dead battery that is being used to make a call should make a request that retains		

R2.0 Assertion #	Description	Test	Comments
	sufficient power to continue the call.		
8.2.5.2.1 Local device handling of mismatch			
8.2.5.2.1#r1	• A more sophisticated Device with a user interface, e.g., a mobile device or monitor, should provide notification through the user interface on the Device.		
8.2.5.2.2 Device Policy Manager Communication with System Policy			
8.2.5.2.2#r1	The System Policy Manager should ensure that the user is informed of the condition.		
8.2.5.2.2#r2	When another port in the system could satisfy the Consumer's power requirements the user should be directed to move the Device to the alternate port.		
8.2.6.2 Battery supplies			
8.2.6.2#r1	When two devices are connected that are not Externally Powered, they should define their own policies so as to prevent constant role swapping.		
8.2.7.4 Device Policy Manager in a Provider/Consumer dead battery handling			
8.2.7.4#r1	The Device Policy Manager in a Provider/Consumer with a battery should also provide: • Detection and handling of back powering in the case of a Type_A to Type_B Dead battery (see Section 4.1.1).		
8.3.3.1 Introduction to state diagrams used in Chapter 8			
8.3.3.1#r1	Conditions listed on state transitions will come from one of three sources and, when there is a conflict, should be serviced in the following order: 1. Message and related indications passed up to the Policy Engine from the Protocol Layer (message sent, message received etc.) 2. Events triggered within the Policy Engine e.g. timer timeouts. 3. Information and requests coming from the Device Policy manager relating either to Local Policy, or to other modules which the Device Policy Manager controls such as Power Supply and Cable Detection.		
8.3.3.2.3 Send Capabilities			
8.3.3.2.3#r1	The Policy Engine should operate as if the device is unattached until such time as a detach/reattach is detected.		
8.3.3.3.6 Transition Sink			
8.3.3.3.6#r1	Note that if there is no power level change the Device Policy Manager should not affect any change to the Power Supply.		
8.3.3.3.7 Ready			
8.3.3.3.7#r1	On entry to the PE_SNK_Ready state as the result of a wait the Policy Engine should do the following: • Initialize and run the SinkRequestTimer.		
8.3.3.3.7#r2	Note this should not cause the SinkRequestTimer to be reinitialized.		
8.3.3.6.10 BFSK Provider/Consumer Dead Battery/Power Loss State Diagram			
8.3.3.6.10#r1	If the Provider/Consumer is powered on and has sufficient power to power its port it should start up as a Source Port.		

7.8 Chapter 9 System Policy

7.8.1 Normative (Shall/Normative)

R2.0 Assertion #	Description	Test	Comments
Subsection 9.1 Overview			
9.1#1	All PD Capable USB (PDUSB) Devices shall report themselves as self-powered devices (over USB) when plugged into a PD capable port even if they are entirely powered from VBUS.	SYSTEM POLICY SPECIFIC TESTS	
9.1#2	PDUSB Devices shall follow the PD requirements when it comes to suspend (see Section 6.4.1.2.3.2 of the PD specification), configured, and operational power.	USB30CV/USB20CV current measurement tests	
9.1#3	The power requirements reported in the PD Consumer Port descriptor of the device shall override the power draw reported in the bMaxPower field in the configuration descriptor.	USB30CV/USB20CV current measurement tests	
9.1#4	A PDUSB Device shall report zero in the bMaxPower field after successfully negotiating a mutually agreeable power contract	SYSTEM POLICY SPECIFIC TESTS	
9.1#5	A PDUSB Device shall disconnect and re-enumerate when it switches operation back to operating in standard [USB2.0], [USB3.1], [USBType_C1.0] or [BC1.2].		
9.1#6	When operating in [USB2.0], [USB3.1], [USBType_C1.0] or [BC1.2] mode it shall report its power draw via the bMaxPower field.	USBCV tests	
9.1.1 PDUSB Device and Hub Requirements			
9.1.1#1	All PDUSB Devices with data lines shall return all relevant descriptors mentioned in chapter 9 of the PDUSB specification.	SYSTEM POLICY SPECIFIC TESTS	
9.1.1#2	PDUSB Hubs shall also support a PD notification capability as defined in chapter 9 of the PDUSB specification.		
9.1.1#3	PDUSB Hubs shall support the capability to return the PD specific capabilities of a device that is attached to any of its downstream ports.	SYSTEM POLICY SPECIFIC TESTS	
9.1.1#4	PDUSB Hubs shall support Local Policy Mode and shall report the appropriate status information in response to any status requests from the System Policy Manager when operating in Local Policy Mode.		
9.1.2 Mapping to USB Device States			
9.1.2#1	The device shall determine whether or not it is in the USB Attached or USB Powered states as described in Figure 9.3, Figure 9.4 and Figure 9.5 of the PD specification.		
9.1.2#2	All other USB states of the PDUSB Device shall be as described in Chapter 9 of [USB2.0] and [USB3.1].		
9.2 PD Class Specific Descriptors			
9.2#1	A PDUSB Device shall return all relevant descriptors mentioned in this section.	SYSTEM POLICY SPECIFIC TESTS	
9.2#2	The device shall return its capability descriptors as part of the device's Binary Object Store (BOS) descriptor set.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1 USB Power Delivery Capability Descriptor			
9.2.1#1	The bReserved field in USB Power Delivery Capability Descriptor shall be set to zero.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#2	The Reserved bits (0, 7, 31:15) of bmAttributes in USB Power Delivery Capability Descriptor shall be set to zero.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#3	The Battery Charging bit of bmAttributes in USB Power Delivery Capability Descriptor shall be set to one if the device supports Battery Charging, else 0.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#4	The USB Power Delivery bit of bmAttributes in USB Power Delivery Capability Descriptor shall be set to one if the device supports USB Power Delivery.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#5	The Provider bit of bmAttributes in USB Power Delivery Capability Descriptor shall be set to one if the device is capable	SYSTEM POLICY	

R2.0 Assertion #	Description	Test	Comments
	of providing power.	SPECIFIC TESTS	
9.2.1#6	The Consumer bit of bmAttributes in USB Power Delivery Capability Descriptor shall be set to one if the device is a consumer of power.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#7	At least one of the bits 8, 9 or 14 of the bmPowerSource bits of USB Power Delivery Capability Descriptor shall be set to 1.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#8	If a PDUSB peripheral supports an AC power supply, the AC Supply bit of bmAttributes in the USB Power Delivery Capability Descriptor shall be set to 1, else set to 0.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#9	If the PDUSB peripheral supports a battery, the Battery bit of bmAttributes in the USB Power Delivery Capability Descriptor shall be set to 1, else 0.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#10	If the PDUSB device has at least one battery, the NumBatteries bits of bmAttributes in the USB Power Delivery Capability Descriptor shall report the number of batteries in the device in bits 13:11 of the bmAttributes field.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#11	If a PDUSB peripheral uses Vbus, the Uses Vbus bit of bmAttributes in USB Power Delivery Capability Descriptor shall be set to 1, else 0.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#12	The bmProviderPorts field of the USB Power Delivery Capability Descriptor shall contain a bitmap indicating which ports are capable of providing power as described in table 9-2 of the PD specification.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#13	The bmConsumerPorts field of the USB Power Delivery Capability Descriptor shall contain a bitmap indicating which ports are capable of consuming power as described in table 9-2 of the PD specification.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#14	The bcdBCVersion field of the USB Power Delivery Capability Descriptor shall indicate the BC version supported by the device if the device indicates that it supports BC in the bmAttributes field.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#15	The bcdPDVersion field of the USB Power Delivery Capability Descriptor shall indicate the PD version supported by the device if the device indicates that it supports PD in the bmAttributes field.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#16	The bcdUSBTypeCVersion field of the USB Power Delivery Capability Descriptor shall indicate the USB Type-C Cable version supported by the device if the device indicates that it supports USB Type-C in the bmAttributes field	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#17	The bLength field in the USB Power Delivery Capability Descriptor shall be set to 18.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#18	The bDescriptorType field in the USB Power Delivery Capability Descriptor shall be set to 10h.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#19	The bDevCapabilityType field in the USB Power Delivery Capability Descriptor shall be set to 6h.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#20	The CHARGING_POLICY bit (bit 5) of bmAttributes in USB Power Delivery Capability Descriptor shall be set to one if the device supports the CHARGING_POLICY feature.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#21	The Type C Current bit (bit 6) of bmAttributes in USB Power Delivery Capability Descriptor shall be set to one if the device supports power capabilities defined in the USB Type C specification.	SYSTEM POLICY SPECIFIC TESTS	
9.2.1#22	The bcdTypeCVersion field of the USB Power Delivery Capability Descriptor shall indicate the Type C version supported by the device if the device indicates that it supports Type C current in the bmAttributes field.	SYSTEM POLICY SPECIFIC TESTS	
9.2.2 Battery Info Capability Descriptor			

R2.0 Assertion #	**Description**	**Test**	**Comments**
9.2.2#1	A PDUSB Device shall support the Battery Info Capability Descriptor if it reported that one of its power sources was a Battery in the bmPowerSource field in its Power Deliver Capability Descriptor.	SYSTEM POLICY SPECIFIC TESTS	
9.2.2#2	A PDUSB device shall return one Battery Info Descriptor per battery it supports.	SYSTEM POLICY SPECIFIC TESTS	
9.2.2#3	The iBattery field of the Battery Info Capability Descriptor shall contain the index of string descriptor that contains the user friendly name for this battery.	SYSTEM POLICY SPECIFIC TESTS	
9.2.2#4	The iSerial field of the Battery Info Capability Descriptor shall contain the index of the string descriptor that contains the Serial Number String for this battery.	SYSTEM POLICY SPECIFIC TESTS	
9.2.2#5	The iManufacturer field of the Battery Info Capability Descriptor shall contain the index of string descriptor that contains the name of the Manufacturer for this battery.	SYSTEM POLICY SPECIFIC TESTS	
9.2.2#6	The bBatteryID field of the Battery Info Capability Descriptor shall be used to uniquely identify this battery in status messages.	not tested	We could at least test whether the bBatteryID field is unique on the device.
9.2.2#7	The bReserved field of the Battery Info Capability Descriptor shall be set to zero.	SYSTEM POLICY SPECIFIC TESTS	
9.2.2#8	The dwChargedThreshold of the Battery Info Capability Descriptor shall contain the Battery Charge value above which this battery is considered to be fully charged but not necessarily “topped off.”		
9.2.2#9	The dwWeakThreshold field of the Battery Info Capability Descriptor shall contain the minimum charge level of this battery such that above this threshold, a device can be assured of being able to power up successfully.		
9.2.2#10	The dwBatteryDesignCapacity field of the Battery Info Capability Descriptor shall contain the design capacity of the battery.		
9.2.2#11	The dwBatteryLastFullchargeCapacity field of the Battery Info Capability Descriptor shall contain the maximum capacity of the battery when fully charged.		
9.2.2#12	The bLength field in the Battery Info Capability Descriptor shall be set to 24.	SYSTEM POLICY SPECIFIC TESTS	
9.2.2#13	The bDescriptorType field in the Battery Info Capability Descriptor shall be set to 10h.	SYSTEM POLICY SPECIFIC TESTS	
9.2.2#14	The bDevCapabilityType field of the Battery Info Capability Descriptor shall be set to 7h.	SYSTEM POLICY SPECIFIC TESTS	
9.2.3 PD Consumer Port Capability Descriptor			
9.2.3#1	A PDUSB Device shall support this capability descriptor if it reported that one or more of its ports is a Consumer ports, as described in the bmConsumerPorts field in its Power Deliver Capability Descriptor.	SYSTEM POLICY SPECIFIC TESTS	
9.2.3#2	A PDUSB device shall return one PD Consumer Port Capability descriptor per port that is a Consumer.	SYSTEM POLICY SPECIFIC TESTS	
9.2.3#3	A PDUSB device shall return PD Consumer Port Capability Descriptors in port order number (low to high).	Not tested.	Not testable
9.2.3#4	A PDUSB Peripheral Device shall have at most one Consumer or Consumer/Provider port.	SYSTEM POLICY SPECIFIC TESTS	
9.2.3#5	A PDUSB Hub shall have a maximum of sixteen Consumer capable ports (Consumer, Consumer/Provider or	SYSTEM POLICY SPECIFIC TESTS	

R2.0 Assertion #	Description	Test	Comments
	Provider/Consumer.		
9.2.3#6	The bReserved field of the PD Consumer Port Descriptor shall be set to zero.	SYSTEM POLICY SPECIFIC TESTS	
9.2.3#7	Bits 15:3 of the bmCapabilities field shall be set to zero.	SYSTEM POLICY SPECIFIC TESTS	
9.2.3#8	The wMinVoltage field of the PD Consumer Port Descriptor shall contain the minimum voltage in 50mV units that this Consumer is capable of operating at.	not tested	
9.2.3#9	The wMaxVoltage field of the PD Consumer Port Descriptor shall contain the maximum voltage in 50mV units that this Consumer is capable of operating at.	not tested	
9.2.3#10	The wReserved field of the PD Consumer Port Descriptor shall be set to zero.	SYSTEM POLICY SPECIFIC TESTS	
9.2.3#11	The dwMaxOperatingPower field of the PD Consumer Port Descriptor shall contain the maximum power in 10mW units this Consumer can draw when it is in a steady state operating mode.	modify existing current measurement tests	
9.2.3#12	The dwMaxPeakPower field of the PD Consumer Port Descriptor shall contain the maximum power in 10mW units this Consumer can draw for a short duration of time (dwMaxPeakPowerTime) before it falls back into a steady state.	modify existing current measurement tests	
9.2.3#13	The dwMaxPeakPowerTime field of the PD Consumer Port Descriptor shall contain the time in 100ms units that this Consumer can draw peak current.	not tested	
9.2.3#14	The dwMaxPeakPowerTime field of the PD Consumer Port Descriptor shall be set to 0xFFFF if this value is unknown.	not tested	
9.2.3#15	The bLength field of the PD Consumer Port Descriptor shall be set to 24.	SYSTEM POLICY SPECIFIC TESTS	
9.2.3#16	The bDescriptorType field of the PD Consumer Port Descriptor shall be set to 10h.	SYSTEM POLICY SPECIFIC TESTS	
9.2.3#17	The bDevCapability field of the PD Consumer Port Capability Descriptor shall be set to 8h	SYSTEM POLICY SPECIFIC TESTS	
9.2.3#18	The bmCapabilities USB Power Delivery bit of the PD Consumer Port Descriptor shall be set if the consumer port supports Power Delivery.	SYSTEM POLICY SPECIFIC TESTS	
9.2.4 PD Provider Port Capability Descriptor			
9.2.4#1	A PDUSB Device shall support the PD Port Capability Descriptor if it reported that one or more of its ports is a Provider port, as described in the bmProviderPorts field in its Power Deliver Capability Descriptor.	SYSTEM POLICY SPECIFIC TESTS	
9.2.4#2	A PDUSB device shall return one PD Provider Port Capability Descriptor per port that is a Provider.	SYSTEM POLICY SPECIFIC TESTS	
9.2.4#3	A PDUSB device shall return PD Port Provider Capability Descriptors in port order number (low to high).	Not tested	Not testable in CV.
9.2.4#4	A PDUSB Peripheral Device shall have at most one Provider port.	SYSTEM POLICY SPECIFIC TESTS	
9.2.4#5	A PDUSB Hub shall have a maximum of sixteen Provider ports.	SYSTEM POLICY SPECIFIC TESTS	
9.2.4#6	The bReserved fields of the PD Provider Port Capability Descriptor shall be set to zero.	SYSTEM POLICY SPECIFIC TESTS	
9.2.4#7	The bmCapabilities Battery Charging bit of the PD Provider Port Capability Descriptor shall be set if the device supports Battery Charging.		
9.2.4#8	The bNumOfPDObjects field of the PD Provider Port Descriptor shall indicate the number of Power Data Objects.	SYSTEM POLICY	

R2.0 Assertion #	Description	Test	Comments
		SPECIFIC TESTS	
9.2.4#9	The PD Provider Port Descriptor shall contain bNumOfPDObjects Power Data Objects starting at wPowerDataObject1 field.		
9.2.4#10	All Power Data Objects contained in a PD Provider Port Descriptor shall be 4 bytes long.		
9.2.4#11	The bLength field of the PD Provider Port Descriptor shall be set to 8 + (4 * bNumOfPDObjects).		
9.2.4#12	The bDescriptorType field of the PD Provider Port Descriptor shall be set to 10h	SYSTEM POLICY SPECIFIC TESTS	
9.2.4#13	The bDevCapabilityType field of the PD Provider Port Descriptor shall be set to 9h.	SYSTEM POLICY SPECIFIC TESTS	
9.2.4#14	The bmCapabilities USB Power Delivery bit of the PD Provider Port Capability Descriptor shall be set if the device supports Power Delivery.	SYSTEM POLICY SPECIFIC TESTS	
9.2.4#15	Bits 15:3 of bmCapabilities field of PD Provider Port Descriptor shall be set to 0.	SYSTEM POLICY SPECIFIC TESTS	
9.3 PD Class Specific Requests and Events			
9.3#1	A PDUSB Hub shall support the PD specific requests and events detailed in section 9.3.1 of the PD specification whether the PDUSB Hub is a Power Provider, a Power Consumer, or both.		
9.3#2	A PDUSB Device that is compliant to this specification shall support the battery related requests if it has a battery.		
9.3#3	PDUSB hub events shall be sent to the system software in response to changes on the PDUSB Hub ports that are not a direct result of a request from system software.		
9.3#4	PDUSB hub notifications shall be sent over the Interrupt endpoint of a PDUSB Hub.		
9.3#5	A PDUSB Hub shall only send PDUSB notifications, or respond to PD requests (except for GetBatteryStatus) if it has been made aware that the SPM is active on the host as detailed in Section 9.4.4 of the PD specification.		
9.3.1 Class specific Requests			
9.3.1#1	The PD class defines requests to which PDUSB Devices shall respond as outlined in Table 9-6 of the PD specification.		
9.3.1#2	All valid requests in Table 9-6 of the PD specification shall be implemented by PDUSB devices.		
9.3.2 PDUSB Hub Event Reporting			
9.3.2#1	PDUSB Hubs shall report events back to the system via the “Port N change detected” bit in the PDUSB Hub Status Change bitmap returned via the PDUSB Hub notification endpoint.		
9.3.2#2	In order to indicate that a PD change occurred, a PDUSB Hub compliant to the PD class shall set Bit 15 “C_PORT_PD_CHANGE” in the Port Change field when queried using the standard Get Port Status request.		
9.3.2#3	A PDUSB Hub shall send PDUSB hub notifications for all downstream ports irrespective of whether it is a Consumer or Provider on that port.		
9.4.1 Clear Port PD Feature (PDUSB Hub)			
9.4.1#1	If the feature selector is associated with a status change, the Clear Change Mask shall define the status changes that are being acknowledged by the SPM.		
9.4.1#2	Only when all the status changes in PD Status Change field (see Table 9-11 of the PD specification) are acknowledged shall the hub clear the C_PORT_PD_CHANGE bit in the port’s Port Change field.		
9.4.1#3	It is a Request Error if wValue is not a feature selector listed in Table 9-8 of the PD specification that shall apply to a port as		

R2.0 Assertion #	Description	Test	Comments
	recipient, or if wIndex specifies a port that does not exist, or if wLength is not as specified in section 9.4.1 of the PD specification.		
9.4.1#4	When a PD status change is acknowledged using ClearFeature() the corresponding bit shall be set to 0 in the PD status.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2 Get Battery Status (PDUSB Hub/Peripheral Device)			
9.4.2#1	The PDUSB Hub/Peripheral Device shall return the Battery Status of the Battery identified by the value of wIndex field in a Get Battery Status request.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#2	Every PDUSB Device that has a battery shall return its Battery Status when queried with a Get Battery Status request.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#3	A GetBatteryStatus with wValue of 0 and wIndex of a valid Battery ID shall return 8 bytes of data.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#4	For Providers or Consumers with multiple batteries, the status of each battery shall be reported per battery.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#5	The bBatteryAttributes field of data returned from GetBatteryStatus request shall be set to 0 if there is no battery currently attached.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#6	The bBatteryAttributes of data returned from GetBatteryStatus request shall not have a value greater than 3.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#7	The bBatterySOC field of data returned from GetBatteryStatus request shall indicate the Battery State of Charge given as percentage value from Battery Remaining Capacity.	Not tested	
9.4.2#8	The bBatteryStatus field of data returned from GetBatteryStatus request shall indicate the present status of the battery as described in table 9-10 of the PD specification.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#9	The bBatteryStatus field of data returned from GetBatteryStatus request shall not be greater than 7.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#10	The bRemoteWakeCapStatus field of data returned from GetBatteryStatus request shall indicate if the device is enabled for Battery Remote wake events as described in Table 9-10 of the PD specification.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#11	If a peripheral supports remote wake, it shall support Battery remote wake.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#12	The bRemoteWakeCapStatus field of data returned from GetBatteryStatus request shall be set to 0 by default.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#13	Bits 7:3 of the bRemoteWakeCapStatus field of data returned from GetBatteryStatus are reserved and shall be set to 0.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#14	The wRemainingOperatingTime field of data returned from GetBatteryStatus request shall contain the operating time (in minutes) until the Weak Battery threshold is reached.	Not tested	
9.4.2#15	The wRemainingOperatingTime field of GetBatteryStatus request shall exclude any additional power received from charging.	Not tested	
9.4.2#16	The wRemainingOperatingTime field of data returned from GetBatteryStatus request shall return a value of 0xFFFF if the battery is not capable of returning this information.	Not tested	
9.4.2#17	The wRemainingChargeTime field of data returned from GetBatteryStatus shall contain the remaining time (in minutes) until the Charged Battery threshold is reached.	Not tested	
9.4.2#18	The wRemainingChargeTime field of data returned from GetBatteryStatus request shall only be valid if the Charging Flow is “Charging”.	Not tested	
9.4.2#19	The wRemainingChargeTime field of data returned from GetBatteryStatus request shall return a value of 0xFFFF if the	Not tested	

R2.0 Assertion #	Description	Test	Comments
	battery is not capable of returning this information.		
9.4.2#20	If wIndex refers to a Battery that does not exist, then the PDUSB Device shall respond with a Request Error.	Not tested	
9.4.2#21	The bBatteryAttributes field of data returned from GetBatteryStatus request shall be set to 1 if the battery is charging.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#22	The bBatteryAttributes field of data returned from GetBatteryStatus request shall be set to 2 if the battery is discharging.	SYSTEM POLICY SPECIFIC TESTS	
9.4.2#23	The bBatteryAttributes field of data returned from GetBatteryStatus request shall be set to 3 if the battery is neither discharging nor charging.	SYSTEM POLICY SPECIFIC TESTS	
9.4.3 Get Port Partner PD Objects (PDUSB Hub)			
9.4.3#1	When a GET_PARTNER_PDO request is received by the PDUSB Hub it shall return the Consumer PDOs (if wValue is set to zero) or Provider PDOs (if wValue is set to one) for the PDUSB Device connected on the downstream port indicated by the wIndex field.		
9.4.3#2	The PDUSB Hub shall use PD mechanisms (see Section 6.3.8 of the PD specification) to retrieve the Power Data Objects from the PDUSB Device.	Not tested	
9.4.3#3	The PDUSB hub shall return wLength bytes in response to a GET_PARTNER_PDO request.	SYSTEM POLICY SPECIFIC TESTS	
9.4.3#4	If the device on the Partner Port does not support PD, then the PDUSB Hub shall respond with a Request Error.	Not tested	
9.4.4 Get Port PD Status (PDUSB Hub)			
9.4.4#1	A GetPDStatus() request to a PDUSB Hub shall return the 8 byte Port PD Status of the port indicated by wIndex.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#2	A PDUSB hub shall set the PD Status Change bits of Port PD Status to indicate port changes as indicated in Table 9-11 of the PD specification.		
9.4.4#3	The Reserved bits of thePort PD Status shall be set to 0.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#4	When the External supply bit of Port PD status is set, the Supply bit shall indicate the current status of the supply.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#5	When the Port operation mode bit of Port PD Status is set, the Port Operation Mode bits shall indicate the current operational mode of the port.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#6	When the Supported Provider Capabilities bits of Port PD Status is set, Get PD descriptor will return details on the notification.		
9.4.4#7	When the Negotiated power level bits of Port PD Status is set and the PDUSB hub is operating in non-intrusive mode, the Request Data Object field shall indicated the newly negotiated power level.		
9.4.4#8	When the Negotiated power level bit of Port PD Status is set and the PDUSB Hub is working in intrusive mode, the Request Data Object field shall indicate the power level that the port partner wants to operate at.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#9	A PDUSB Hub operating in Non-Intrusive Mode shall the New Power Direction bit of Port PD Status after it has changed power direction.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#10	A PDUSB Hub operating in Intrusive Mode shall set the New Power Direction bit of Port PD Status if the port partner wants to perform a power swap.		
9.4.4#11	A PDUSB Hub shall set the PD Reset Complete bit of Port PD Status after a completed PD Hard Reset.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#12	When the Reset Complete bit of Port PD Status is set, all other	SYSTEM POLICY	

R2.0 Assertion #	Description	Test	Comments
	PD Status Change bit fields shall report their change status as if the PDUSB Hub was working in Non-Intrusive mode.	SPECIFIC TESTS	
9.4.4#13	The Insertion Detect bit of the Port PD Status shall indicate the current status of whether a cable is plugged into this port.		
9.4.4#14	The PD Capable Cable bit of Port PD Status shall indicate the current status of whether a PD Capable cable is plugged into this port.		
9.4.4#15	The Request Rejected bit of the Port PD Status shall only be valid if the PDUSB Hub is operating in Non-Intrusive mode.		
9.4.4#16	The Request Rejected bit of Port PD Status shall be set when a PDUSB Hub operating in Non-Intrusive Mode rejects a power level request.		
9.4.4#17	The Request Rejected bit of Port PD Status shall be set when a PDUSB Hub operating in Non-Intrusive Mode rejects a role swap request.		
9.4.4#18	The HybridMode Request bit of Port PD Status shall only be set by the PDUSB Hub when it is operating in Hybrid mode.		
9.4.4#19	When the HybridMode Request bit of Port PD Status is set, then the Negotiated Power Level and New Power Direction bits of Port PD Status shall be interpreted as if the hub was operating in Intrusive mode.		
9.4.4#20	The Power Role Swap Completed bit of the Port PD Status shall be set when the hub completes a Power Role Swap requested by the SPM.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#21	The Error bit of Port PD Status shall indicate that an unknown error has occurred on the port.	not tested	
9.4.4#22	The Supply bit of Port PD Status shall be set to 0 if no supply is present, else 1.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#23	The Supply bit of Port PD Status shall have the same value as Externally Powered bit in Source/Sink Capability message.	not tested	
9.4.4#24	The Port Operation Mode bits of Port PD Status change is set, the Port Operation Mode bits shall indicate the current mode of operation of the port as described in Table 9-11 of USB PD spec.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#25	The Insertion Detect bit of Port PD Status shall be set to 1 if a cable is detected on the port, else 0.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#26	The PD Capable Cable bit of Port PD Status shall be set to 1 if a PD capable cable is detected on this port, else 0.	SYSTEM POLICY SPECIFIC TESTS	
9.4.4#27	The PD Direction bit of Port PD Status shall be set to 0 if this port is operating as a consumer else 1.		
9.4.4#28	The Request Data Object field of Port PD Status shall return the currently negotiated power level if the hub was operating in Non-Intrusive mode.		
9.4.4#29	The Request Data Object field of Port PD Status shall return the power level that the port partner wants to operate at if the PDUSB Hub is working in intrusive mode.		
9.4.4#30	If the Request Rejected bit is set because of a rejected power negotiation, the Negotiated power level bit shall also be set.		
9.4.4#31	If the Request Rejected bit is set because of a rejected swap, the new power direction bit shall also be set.		
9.4.5.1 BATTERY_WAKE_MASK Feature Selector			
9.4.5.1#1	When the Battery Present bit of the wIndex field is set then the PDUSB Device shall generate a wake event if it detects that a battery has been attached.		
9.4.5.1#2	When the Charging Flow bit of the wIndex field is set then the PDUSB Device shall generate a wake event if it detects that a battery switched from charging to discharging or vice versa.		
9.4.5.1#3	When the Battery Error bit of the wIndex field is set then the		

R2.0 Assertion #	Description	Test	Comments
	PDUSB Device shall generate a wake event if the battery has detected an error condition.		
9.4.5.1#4	Bits 15:3 of the wIndex field are reserved and shall not be used.		
9.4.5.2 OS_IS_PD_AWARE Feature Selector			
9.4.5.2#1	The default state for the OS_IS_PD_AWARE feature shall be zero.		
9.4.5.2#2	The PDUSB Hub shall respond to all PD requests (except GetBatteryStatus) or send any PD notifications only after the OS_IS_PD_AWARE feature is set.		
9.4.5.3 POLICY_MODE Feature Selector			
9.4.5.3#1	When the feature selector is set to POLICY_MODE, the wIndex field shall be set to one of the values in Table 9-13 of the PD specification.		
9.4.5.3#2	When the feature selector is set to POLICY_MODE, values 03H-FFFFH of the wIndex field are reserved and shall not be used.		
9.4.5.3#3	If the Policy Mode is set to intrusive, then the PDUSB Hub shall be set to operate in Intrusive Mode.		
9.4.5.3#4	If the Policy Mode is set to Local or Hybrid, then the PDUSB hub shall be set to operate in Non-Intrusive mode.		
9.4.5.3#5	When the Policy Mode is set to Intrusive then the PDUSB Hub shall pass any requests that need policy decisions to be made to the SPM		
9.4.5.3#6	When the Policy Mode is set to Intrusive then the PDUSB Hub shall use PD mechanisms to delay the completion of PD Requests (e.g. using the Wait Message) over PD.		
9.4.5.3#7	When the Policy Mode is set to Hybrid then only requests which cannot be accommodated by the PDUSB Hub shall be sent to the SPM.		
9.4.5.3#8	It is shall be a Request Error if wValue is not a feature selector listed in Table 9-8 of the PD specification.	Not tested	
9.4.5.3#9	It is shall be a Request Error if wIndex is not set to a value as specified above.	Not tested	
9.4.5.4 CHARGING_POLICY Feature Selector			
9.4.5.4#1	When the feature selector is set to CHARGING_POLICY, the wIndex field shall be set to one of the values defined in Table 9-14 of the PD specification.		
9.4.5.4#2	If the device is using USB Type-C Current above the default value or is using PD then this feature setting has no effect and the rules for power levels specified in the [USBType_C1.0] or USB PD specifications shall apply.		
9.4.5.4#3	If the Charging Policy Encoding is set to 00H, the device shall follow the default current limits as defined in the USB 2.0 or USB 3.1 specification, or as negotiated through other USB mechanisms such as BC.		
9.4.5.4#4	If the Charging Policy Encoding is set to 01h, the device shall limit its current draw to the higher of these two values: (1) ICCHPF as defined in the USB 2.0 or USB 3.1 specification, regardless of its USB state; or (2) Current limit as negotiated through other USB mechanisms such as BC.		
9.4.5.4#5	If the Charging Policy Encoding is set to 02h, the device shall limit its current draw to the higher of these two values: (1) ICCLPF as defined in the USB 2.0 or USB 3.1 specification, regardless of its USB state; or (2) Current limit as negotiated through other USB mechanisms such as BC.		
9.4.5.4#6	If the Charging Policy Encoding is set to 03H, the device shall not consume any current for charging the device itself regardless of its USB state.		
9.4.5.4#7	Charging Policy Encodings 04H-FFFFH are reserved and shall not be used.		

R2.0 Assertion #	Description	Test	Comments
9.4.6 Set Port PD Feature (PDUSB Hub)			
9.4.6#1	The SetPortPDFeature request shall only be valid when the PDUSB Hub is operating in Intrusive mode.	Not tested	
9.4.6#2	If the PDUSB Hub cannot satisfy a SetPortPDFeature request then it shall respond with a Request Error to the SPM.	not tested	
9.4.6#3	The wIndex field shall indicate the PD capable port that is being queried.	not tested	
9.4.6#4	The port number shall be a valid port number for that hub, greater than zero.	not tested	
9.4.6#5	When operating in intrusive mode the PDUSB Hub shall respond with a Wait message when it receives any Request message.		
9.4.6#6	When the feature selector is set to SWAP then the PDUSB Hub shall initiate a Role Swap with the port partner on the specified port.		
9.4.6#7	When the feature selector is set to GOTO_MIN and the downstream port is operating as a Provider, then the PDUSB Hub shall send the GotoMin request to the port partner on the port specified in wIndex.		
9.4.6#8	When the feature selector is set to RETURN_POWER and the downstream port is operating as a Provider, then the PDUSB Hub shall return the power that it borrowed from the PD Device on port specified in wIndex.	SYSTEM POLICY SPECIFIC TESTS	
9.4.6#9	When the SetPortPDFeature() request feature selector is set to ACCEPT_PD_REQUEST then the PDUSB Hub shall accept the request that was previously made by the port partner on this downstream port.		
9.4.6#10	When the SetPortPDFeature() request feature selector is set to REJECT_PD_REQUEST then the PDUSB Hub shall reject the request that was previously made by the port partner on this downstream port.		
9.4.6#11	When the SetPortPDFeature() request feature selector is set to PORT_PD_RESET, the port shall be sent a Hard Reset.		
9.4.6#12	Once the Hard Reset initiated by SetPortPDFeature(PORT_PD_RESET) is complete, as described in Section 8.3.2.8 of the PD specification, the hub shall set the C_PORT_PD_CHANGE bit in the port change field.	SYSTEM POLICY SPECIFIC TESTS	
9.4.6#13	When the feature selector is set to CABLE_PD_RESET, the hub shall send a Cable Reset to the cable attached on that Port. Once the Cable Reset is complete, as described in Section 6.7.3 of the PD specification, the hub shall set the C_PORT_PD_CHANGE bit in the Port change field.		
9.4.6#14	It shall be a Request Error if wValue is not a feature selector listed in Table 9-8, or if wIndex specifies a port that does not exist or if the port is operating as a Consumer, or if wLength is not as specified above.	Not tested	
9.4.7 Set Port PDOs (PDUSB Hub)			
9.4.7#1	A SetPortPDOs() request shall cause the PDUSB hub Provider to send an updated Capabilities message.	SYSTEM POLICY SPECIFIC TESTS	
9.4.7#2	The PDOs sent after a SetPortPDOs() request shall constitute a subset of the capabilities supported by the PDUSB Hub as defined in the PD Provider Port Capability Descriptor (see Section 9.2.4 of the PD specification).	SYSTEM POLICY SPECIFIC TESTS	
9.4.7#3	When any PDO set specified by a SetPortPDOs request exceeds the capabilities of the PDUSB Hub the hub shall reject the request.	SYSTEM POLICY SPECIFIC TESTS	
9.4.7#4	The wIndex field shall indicate the PD capable port that is being queried.	SYSTEM POLICY SPECIFIC TESTS	

R2.0 Assertion #	Description	Test	Comments
9.4.7#5	The port number shall be a valid port number for that hub, greater than zero.	not tested	
9.4.7#6	The wLength field shall be a multiple of 4.	not tested	
9.4.7#7	The wLength field shall include all the PDOs (see Section 6.4.1 of the PD specification) that the port may send to its port partner when sending its PD capabilities.	not tested	
9.4.7#8	The port shall send the PDOs in the order that they are included in this request.		
9.4.7#9	A PDUSB Hub shall respond with a Request Error if the PDOs sent by the SPM are not valid for the port specified.		
9.4.7#10	It shall be a Request Error if wValue is not set to zero or if wIndex specifies a port that does not exist, or if wLength is not as specified above.	SYSTEM POLICY SPECIFIC TESTS	
9.4.8 Get VDMs (PDUSB Hub)			
9.4.8#1	The response to each request shall contain only one VDM.		
9.4.8#2	VDMs shall be returned in the order they were received by the Port.		
9.4.8#3	If there are no VDMs to return for the Port the PDUSB Hub shall return a zero length packet.		
9.4.8#4	The wIndex field shall indicate the PD capable Port that is being queried.		
9.4.8#5	The Port number shall be a valid Port number for that PDUSB hub, greater than zero.		
9.4.8#6	The wLength field shall be a multiple of 4 and shall include all the VDOs that the PDUSB Hub received on that Port. (See Section 6.4.4 of the PD specification.)		
9.4.8#7	It shall be a Request Error if wIndex specifies a Port that does not exist, or if wValue or wLength is not as specified above.		
9.4.9 Send VDMs (PDUSB Hub)			
9.4.9#1	The wIndex field shall indicate the PD capable Port that is being queried.		
9.4.9#2	The Port number shall be a valid Port number for that PDUSB hub, greater than zero.		
9.4.9#3	If the wValue field is 0, it shall be sent to its Port partner.		
9.4.9#4	If the wValue field is 1, it shall be sent to SOP'.		
9.4.9#5	If the wValue field is 2, it shall be sent to SOP".		
9.4.9#6	The wLength field shall be a multiple of 4 and shall include one VDM (see Section 6.4.4 of the PD specification) that the Port shall send to the recipient indicated by wValue.		
9.4.9#7	The PD USB shall send the structured VDMs in the order that they are included in this request.		
9.4.9#8	It shall be a Request Error if wIndex specifies a Port that does not exist, or if wValue or wLength is not as specified above.		

7.8.2 Recommended (Should)

R2.0 Assertion #	Description	Test	Comments
Subsection 9.1.3 PDUSB Device Enumeration			
9.1.3#r1	If a PDUSB Device cannot perform its intended function with the amount of power that it can get from the port it is connected to then the host system should display a message (on a PD aware OS) about the failure to provide sufficient power to the device.		

7.9 Appendix A Power Profiles

7.9.1 Recommended (Should)

R2.0 Assertion #	Description	Test	Comments
Subsection A.4.1 Notebook and HDD Examples			
A.4.1#r1	In the case of the HDD with separate power adapter plugged in, the HDD should not draw power from the notebook's port.		

7.10 Appendix C Power Implementation Considerations

7.10.1 Recommended (Should)

R2.0 Assertion #	Description	Test	Comments
Subsection C Power Implementation Considerations			
C#r1	Many of these may not have been as important in legacy USB systems, but should be considered when implementing the PD specification.		
C.1.2 Spurious Noise Test Setup and Calibration			
C.1.2#r1	During calibration the power converter and load impedances should be in place with the power converter turned off.		
C.1.2#r2	Noise levels should be validated across expected line and load conditions including expected combinations of USB Cable types and lengths.		
C.2 Connector Detach Transients			
C.2#r1	Clamping elements should be included or the transceiver should be capable of absorbing the energy of the attach and detach events to prevent damage to the transceiver.		
C.2#r2	An external or integrated clamp should be implemented to absorb this energy and limit the applied voltage at the transceiver input.		
C.3.1 Basic Power Stage Small Signal AC Model			
C.3.1#r1	Figure C 6 shows a simplified diagram of the small signal power stage AC response including the parasitic elements that should be considered.		

7.11 Appendix G VDM Command Examples

7.11.1 Normative (Shall/Normative)

R2.0 Assertion #	Description	Test	Comments
Subsection G Enter Mode Example			
G.3#1	The Responder that is the target of the Enter Mode Command request shall send a Message Header with the Number of Data Objects field set to a value of 1 followed by a VDM Header with the Command Source (B5) set to 1 indicating the command is from a Responder and the Command (B4..0) set to 4 indicating the Responder has entered the Mode and is ready to operate.		

8 PARAMETERS, TIMERS AND COUNTERS

Parameter	Description	Test	Comments
SECTION 3 Cable Assemblies and Connectors			
Table 3-7 Electrical Parameters			
aInsertionLoss	Insertion loss in the cable as measured in fRange		
cPlug	Low Power PD Micro-A plug marking		
iCable	Rate current of the cable either 3 or 5A		
rID	PD Micro-A plug marking		
vIRDrop_Cable	Sum of V_{VBUS_DROP} and V_{GND_DROP}.		
V_{GND_DROP}	Voltage drop on GND as shown in Figure 3-27.		
V_{VBUS_DROP}	Voltage drop on V_{BUS} as shown in Figure 3-27.		
SECTION 4 Electrical Requirements			
Table 4-3 Electrical Parameters			
pLowPower	Maximum power used by a Low Power device	not tested	
tBitStreamDetect	Time valued used for BitStreamDetectTimer	POW-DB-CP	Used in POW-DB-PC
tBitStreamOff	Time for Bit Stream to stop on the appearance of vSafe0V on V_{BUS}.	PHYFSK-DB POW-DB-CP	
tDBDetect	Time value used for DBDetectTimer	POW-DB-CP	Used in POW-DB-PC
tDBDischargeVbus	Time allowed to discharge V_{BUS} to vSafe0V during Dead Battery detection.	POW-DB-CP	
tDBSourceOff	Time value used for DBSourceOffTimer	POW-DB-CP	
tDeviceReady	Time value used by DeviceReadyTimer	PHYFSK-DB	
tSendBitStream	Time to start Bit Stream after the appearance of vSafeDB on V_{BUS}.	PHYFSK-DB POW-DB-PC	
tWaitForPower	Time value used by WaitForPowerTimer	PHYFSK-DB	
vLowPower	Voltage range for operation of Low Power devices	not tested	
Table 4-4 Electrical Timers			
BitStreamDetectTimer	Used to determine how long to wait for the Bit Stream	not parameter	
DBDetectTimer	Used to determine when next to apply vSafeDB	not parameter	
DBSourceOffTimer	Used to determine the time from driving V_{BUS} to vSafe0V and starting the DBDetectTimer.	not parameter	
DeviceReadyTimer	Used to ensure that the Provider/Consumer is ready to operate as a Sink within a reasonable time during Dead Battery operation	not parameter	
WaitForPowerTimer	Used to trigger bringing up of the	not parameter	
SECTION 5 Physical Layer			
Table 5-9 Common Normative Requirements			
fCarrier	FSK Carrier Frequency	PHYFSK-TX-FREQ PHY_DB	
fDeviation	FSK frequency deviation	PHYFSK-TX-FREQ PHY_DB	
fBitRate	Bit Rate	PHY_TX_BIT PHY_DB	
Table 5-10 Transceiver Isolation Impedance Normative Requirements			

Parameter	Description	Test	Comments
zIsolation	Impedance Allowed	PHYFSK-TERM	
fRange	Range of frequencies used in communication	more a definition - so not tested	
Table 5-11 Transmitter Requirements			
pBitRate	Maximum difference between the bit-rate during the part of the packet following the preamble and the reference bit-rate.	PHY_TX_BIT	
pCarrierFreq	Maximum difference between the carrier-frequency during the part of the packet following the preamble and the reference carrier-frequency.	PHY_TX_FREQ PHY_RX_BER	
pDevFreq	Maximum difference between the deviation frequency during the part of the packet following the preamble and the reference deviation frequency.	PHY_TX_FREQ PHY_RX_BER	
rTX	The termination resistance and the cable impedance for test and calculation.	PHYFSK-TERM	
tBusIdle	Transceiver turn around time, do not transmit during this time after receiving or transmitting the last bit of a message.	PROT-PROC-GOODCRC-UUT	
vTX	TX voltage injected on V_{BUS}	PHYFSK-TX-LEV PHY_RX_BER	
Table 5-13 Receiver Normative Requirements			
nBER	Bit error rate, S/N = 25 dB		
tBitStreamComplete	The Bit Stream has stopped if the squelch has closed for tBitstreamComplete.	Should be in PHY_DB or POW-DB-PC	
vRX	RX voltage received on V_{BUS}	PHY_TX_LEV* PHY_RX_BER	*captive cable case
vSquelchDetecting	Squelch detection sensitivity in Squelch Detection Mode	not tested	
vSquelchOperating	Squelch detection sensitivity in Squelch Operating Mode	PHYFSK-RX-SQU	
zRX	Receiver input impedance	not tested	
SECTION 6 Protocol Layer			
Table 6-20 Time Values			
tBISTMode	Maximum time that a UUT has to enter a BIST mode		
tBISTReceive	Time that the sender's Protocol Layer has during BIST operation to detect when a frame has been lost and to trigger the transmission of the next test frame		
tBISTResponse	The maximum time which a UUT has to respond with a Test Frame when operating in BIST Transmit Mode		
tHardReset	Time to initiate a Hard Reset after a Soft Reset failure.	PROT-CALC-HR	
tNoResponse	Time value used by NoResponseTimer.	PROT-SR-HR	

Parameter	Description	Test	Comments
tPSSourceOff	Time value used by PSSourceOffTimer.	POW-SWAP-PC-SRC-REQ POW-SWAP-PC-SNK-SRC POW-SWAP-PC-SNK-SNK POW-SWAP-CP-SRC-SRC POW-SWAP-CP-SRC-SNK POW-SWAP-CP-SNK-SRC POW-SWAP-CP-SNK-SNK PROT-PROC-PSSOURCEOFFTIMER PROT-PROC-PSSOURCEOFFTIMER-SWPD	
tPSSourceOn	Time value used by PSSourceOnTimer.	POW-SWAP-PC-SRC-REQ POW-SWAP-PC-SNK-SRC POW-SWAP-PC-SNK-SNK POW-SWAP-CP-SRC-SRC POW-SWAP-CP-SRC-SNK POW-SWAP-CP-SNK-SRC POW-SWAP-CP-SNK-SNK PROT-PROC-PSSOURCEONTIMER PROT-PROC-PSSOURCEONTIMER-SWPD	
tPSTransition	Time value used by PSTransitionTimer.	PROT-HS-PSTRANS-C-CP POW-SRC-LOAD-P-PC POW-SRC-LOAD-CP-ACC POW-SRC-LOAD-CP-REQ	
tReceive	Time value used by CRCReceiveTimer.	PROT-CALC-HR	
tReceiverResponse	Time allowed to respond to a message.	PROT-SEQ-SWAP-REJ PROT-PROC-SWAP-TSTR-SNK PROT-PROC-SWAP-TSTR-SRC PROT-PROC-REQ-TSTR PROT-PROC-SRCCAPS-TSTR PROT-PROC-GETSRCCAPS-TSTR PROT-PROC-GETSRCCAPS-UUT PROT-PROC-GETSNKCAPS-TSTR	

Parameter	**Description**	**Test**	**Comments**
tRemoveV$_{BUS}$	Time to remove V$_{BUS}$ after receiving unexpected capabilities.	PROT-SRC-SRC	
tRetry	Time allowed to retry a message.	PROT-CALC-HR	
tSenderResponse	Time value used by SenderResponseTimer.	PROT-PROC-GETSRCCAPS-TSTR	
tSendSourceCap	Time value used by SourceCapabilityTimer.	PHYFSK-HR PROT-SR-HR	
tSinkActivity	Time value used by SinkActivityTimer.	PROT-CALC-HR PHYFSK-HR	
tSinkRequest	Time value used by SinkRequestTimer.	POW-SRC-LOAD-P-PC POW-SRC-LOAD-CP-ACC POW-SRC-LOAD-CP-REQ POW-SRC-NOIS-P-PC POW-SRC-NOIS-CP-ACC POW-SRC-NOIS-CP-REQ POW-SRC-TRANS-P-PC POW-SRC-TRANS-CP-ACC POW-SRC-TRANS-CP-REQ	
tSinkWaitCap	Time value used by SinkWaitCapTimer.	PHYFSK-DB PROT-PROC-SWAP-TSTR-SNK PROT-PROC-SWAP-TSTR-SRC PROT-PROC-SWAP-UUT-SNK PROT-PROC-SWAP-UUT-SRC PROT-SR-HR POW-DB-CP	
tSoftReset	Time to issue Soft Reset after communications failure.	PROT-SR-HR	

Parameter	Description	Test	Comments
tSourceActivity	Time value used by SourceActivityTimer.	PROT-CALC-HR PROT-PROC-SWAP-TSTR-SNK PROT-PROC-SWAP-TSTR-SRC PROT-PROC-SWAP-UUT-SNK PROT-PROC-SWAP-UUT-SRC PROT-PROC-REQ-TSTR PROT-PROC-REQ-UUT POW-SRC-LOAD-P-PC POW-SRC-LOAD-CP-ACC POW-SRC-LOAD-CP-REQ POW-SRC-NOIS-P-PC POW-SRC-NOIS-CP-ACC POW-SRC-NOIS-CP-REQ POW-SNK-NOIS-C-CP POW-SNK-NOIS-PC POW-SNK-TRANS-C-CP POW-SNK-TRANS-PC POW-SNK-GOTOMIN	
tTransmit	Time allowed to respond with a GoodCRC message.	PROT-PROC-GOODCRC-UUT PROT-PROC-GETSRCCAPS-TSTR	
Table 6-21 Timers			
BISTReceiveErrorTimer	Used by the sender's Protocol Layer during BIST operation to detect when a frame has been lost and to trigger the transmission of the next test frame		
BISTStartTimer	Used by the Tester to ensure that there is a delay of more than ***tBISTMode***before sending the first Test Pattern Frame after requesting ***BIST Receiver Mode***.		
CRCReceiveTimer	Used by the sender's Protocol Layer to ensure that a message has not been lost.		tReceive
NoResponseTimer	Used by the Policy Engine in a Source or Sink to determine that its Port Partner is not responding after a Hard Reset.		tNoResponse
PSSourceOffTimer	Used by the Policy Engine in Dual-Role Device that is currently acting as a Sink to timeout on a PS_RDY message during a Swap sequence.		tPSSourceOff
PSSourceOnTimer	Used by the Policy Engine in Dual-Role Device that has just stopped sourcing power and is waiting to start sinking power to timeout on a PS_RDY message during a swap.		tPSSourceOn
PSTransitionTimer	Used by the Policy Engine to timeout on a PS_RDY message.		tPSTransition
SenderResponseTimer	Used by the sender's Protocol Layer to ensure that a message requesting a response (e.g. Get_Source_Cap message) is responded to within a bounded time of tSenderResponse.		tSenderResponse
SinkActivityTimer	An absence of a Ping, or other messages within tSinkActivity, as an indication of communications failure		tSinkActivity

Parameter	Description	Test	Comments
SinkRequestTimer	Ensures that the time between Sink Request messages, after a Wait message has been received from the Source, is a minimum of tSinkRequest		tSinkRequest
SinkWaitCapTimer	Used to issue an Hard Reset when there is an absence of Source Capabilities messages.		tSinkWaitCap
SourceActivityTimer	Used to maintain bus activity by periodically sending Ping messages if required.		tSourceActivity
SourceCapabilityTimer	Used to periodically send Source Capabilitues messages.		tSendSourceCap
SwapRecoveryTimer	Used by a Provider/Consumer acting in Sink role during a Hard Reset to wait before turning on its power supply.		tSwapRecover
Table 6-22 Counter Parameters			
nCapsCount	Minimum number of Source Capabilities to be sent before stopping.	PROT-SR-HR	
nHardResetCount	Number of additional hard resets before deciding a device is unresponsive.	PROT-SR-HR	
nMessageIDCount	Maximum value of the MessageIDCounter prior to wrapping.	PROT-HDR	
nRetryCount	Number of retries before generating a Soft Reset.	PROT-SR-HR	
Table 6-23 Counters			
BISTErrorCounter	Used to count the number of bits in error during a BIST Transmit or Receive test	PHYFSK-RX-BER	
CapsCounter	Used to count the number of Source Capabilities sent.	PROT-SR-HR	
HardResetCounter	Used to count the number of resets.	PROT-SR-HR	
MessageIDCounter	A rolling counter, ranging from 0 to nMessageIDCount, used to detect duplicate messages.	PROT-HDR	
RetryCounter	Used to count the number of retries.	PROT-SR-HR	
SECTION 7 Power Supply			
Table 7-23 Source Electrical Parameters			
cSrcBulk	Source bulk capacitance when a port is powered from a dedicated supply.	Vendor Declaration	
cSrcBulkShared	Source bulk capacitance when a port is powered from a shared supply.	Vendor Declaration	
cSrcBulkDB	Source bulk capacitance on V_{BUS} when applying vSafeDB.	Vendor Declaration	
snrSrc	Source's output Signal-to-noise ratio	POW-SRC-NOIS-P-PC POW-SRC-NOIS-CP-ACC POW-SRC-NOIS-CP-REQ POW-SNK-NOIS-C-CP POW-SNK-NOIS-PC	
tNewSnk	Maximum time allowed for an initial Source in Swap Standby to transition new Sink operation.	POW-SNK-TRANS-C-CP POW-SNK-TRANS-PC	
tNewSrcRevertToSink	Hard Reset during Role Swap timing parameter.		
tolSrc	Measured at Source receptacle for Fixed and Programmable supplies only. See note 1.		
tolTranLowerPos	Lower bound of positive transition window.		
tolTranUpperPos	Upper bound of positive transition window.		
tSafeDBtoSafe5V	Transition time from vSafeDB to vSafe5V.		
tShortCctRecover	Source recovery time on fault removal		
tSrcNewPower	Time from positive/negative transition start (t0) to Source ready at new power level		

Parameter	Description	Test	Comments
tSrcPwrRdy	Time alloted for the Source to change its output power capability		
tSrcRecover	Time allotted for the Source to recover		
tSrcSafe	Response time to be at vSafe5V or vSafe0V		
tSrcSettleNeg	Settling time from negative transition start (t0) to when transitioning voltage is within the specified tolerance of the final voltage		
tSrcSettlePos	Settling time from positive transition start (t0) to when transitioning voltage is within the specified tolerance of the final voltage		
tSrcSwapStdby	The mximum time for the Source to transition to Swap Standby.		
tSrcTurnOn	Transition time from vSafe0V to vSafe5V		
tTurnOnSafeDB	Time to turn on vSafeDB source		
vSlewRateNeg	Maximum slew rate allowed for negative voltage transitions. Limits current based on a 3 A connector rating and maximum Sink bulk capacitance of 100 µF.		
vSlewRatePos	Maximum slew rate allowed for positive voltage transitions. Limits current based on a 3 A connector rating and maximum Sink bulk capacitance of 100 µF.		
vSrcNeg	Most negative voltage allowed during transition		
Table 7-24 Sink Electrical Parameters			
cSnkBulk	Sink bulk capacitance on V_{BUS} at attach		
cSnkBulkPd	Bulk capacitance on V_{BUS} a Sink is allowed after a sucessful negotiation		
cSnkBulkDB	Bulk capacitance on V_{BUS} when acting as a Sink during Dead Battery.		
iCapChange	Transient current allowed to flow when the Sink changes its bulk capacitance		
iInRushPos	Current draw rate of change when increasing current		
iInRushNeg	Current draw rate of change when decreasing current		
iSafe0mA	Maximum current a Sink is allowed to draw when V_{BUS} is driven to vSafe0V		
iSnkSwapStdby	Maximum current a Sink can draw during Swap Standby. Ideally this current is very near to 0 mA largely influenced by port leakage current.		
pHubSusp	Suspend power consumption for a hub. 25mW + 25mW per downstream port for up to 4 ports.		
pSnkStdby	Power consumption during Sink Standby		
pSnkSusp	Suspend power consumption for a peripheral device.		
snrSnk	Signal-to-noise ratio of Sink's noise reflected on V_{BUS}.		
tNewSnkRevertToSrc	Hard Reset during Role Swap timing parameter		
tNewSrc	Maximum time allowed for an initial Source in Swap Standby to transition to new Sink operation.		
tSnkHardResetPrepare	Time alloted for the Sink power electronics to prepare for a Hard Reset.		
tSnkNewPower	Maximum transition time between power levels		
tSnkRecover	Time for the Sink to resume default operation.		

Parameter	Description	Test	Comments
tSnkStdby	Time to transition to Sink Standby from Sink		
tSinkTransition	Maximum Time to transition to SwapStandby.		
tSwapRecover	Provides settling time after a hard reset before powering back up.	PROT-PROC-PSSOURCEONTIMER-SWPD PROT-PROC-PSSOURCEONTIMER	
tTurnOnImpliedSink	Turn on time of implied Sink during Dead Battery operation		
Table 7-25 Common Source/Sink Electrical Parameters			
tolTranLowerNeg	Lower bound of negative transition window.		
tolTranUpperNeg	Upper bound of negative transition window.		
vSafe0V	Safe operating voltage at “zero volts”		
vSafe5V	Safe operating voltage at 5V	PROC-PD-MODE PHYFSK-DB PROT-MSG-DATA-SRC-CAP PROT-MSG-DATA-SNK-CAP PROT-PROC-SWAP-TSTR-SNK PROT-PROC-SWAP-TSTR-SRC PROT-PROC-SWAP-UUT-SNK PROT-PROC-SWAP-UUT-SRC PROT-PROC-PSSOURCEOFFTIMER PROT-PROC-PSSOURCEOFFTIMER-SWPD PROT-PROC-PSSOURCEONTIMER PROT-PROC-PSSOURCEONTIMER-SWPD PROT-SRC-SRC PROT-SR-HR POW-SRC-LOAD-P-PC POW-SNK-NOIS-C-CP POW-SNK-NOIS-PC POW-SNK-TRANS-C-CP POW-SWAP-PC-SNK-SNK POW-SWAP-CP-SNK-SRC POW-SWAP-CP-SNK-SNK POW-SNK-GOTOMIN POW-DB-PC POW-DB-CP	
vSafeDB	Safe operating voltage for Dual-Role ports operating as DB Source.	PROC-PD-MODE PHYFSK-DB POW-DB-PC POW-DB-PC	

9 STATE OPERATION V1.3

9.1 Protocol Layer

Assertion	Description	Related Assertion	Comments
SECTION 6.8 State behavior			
Subsection 6.8.2 General Operation			
Subsection 6.8.2.1 Protocol Layer message transmission			
Subsection 6.8.2.1.1 Wait for Message Request State			
	In the Wait for Message Request state the Protocol Layer waits until Policy Engine directs it to send a message. This shall be the startup state for the Protocol Layer message transmission and also the starting state when exiting from a Hard Reset.		
	On entry to the Wait for Message Request state the Protocol Layer shall reset the RetryCounter.	6.7.2#2	
	The Protocol Layer shall transition to the Construct message state when: • A message request is received from the Policy Engine which is not a Soft Reset message.		
	The Protocol Layer shall transition to the Layer Reset for Transmit state when: • A message request is received from the Policy Engine which is a Soft Reset message.		
Subsection 6.8.2.1.2 Layer Reset for Transmit state			
	On entry to the Layer Reset for Transmit state the Protocol Layer shall reset the MessageIDCounter.	6.6.1.1#1a, 6.7.1#2	
	The Protocol Layer shall transition Protocol Layer message reception to the Wait for first PHY Message state (see Section 6.8.2.2.1) in order to reset the stored MessageID.		
	The Protocol Layer shall transition to the Construct Message State when: • The Soft Reset actions in this state have been completed.		
Subsection 6.8.2.1.3 Construct Message State			
	On entry to the Construct Message state the Protocol Layer shall construct the message requested by the Policy Engine, or resend a previously constructed message, and then pass this message to the PHY Layer.		
	The Protocol Layer shall transition to the Wait for PHY Response state when: • The message has been sent to the PHY Layer.		
Subsection 6.8.2.1.4 Wait for PHY Response State			
	On entry to the Wait for PHY Response State, once the message has been sent (see Section 6.5.1), the Protocol Layer shall initialize and run the CRCReceiveTimer.	6.5.1#1, 6.5.1#4	
	The Protocol Layer shall transition to the Match MessageID state when: • A GoodCRC message response is receive from the PHY Layer.	6.5.1#5	
	The Protocol Layer shall transition to the Check RetryCounter state when: • The CRCReceiveTimer times out and • This is not a Soft Reset	6.5.1#2	
	• Or the PHY Layer indicates that a message has been discarded due to V_{BUS} being busy but V_{BUS} is now idle (see Section 5.8).	5.8#5	

Assertion	Description	Related Assertion	Comments
	The Protocol Layer shall transition to the Transmission Error state when: • The CRCReceiveTimer times out and • This is a Soft Reset	6.3.12#2	
Subsection 6.8.2.1.5 Match MessageID State			
	On entry to the Match MessageID state the Protocol Layer shall compare the MessageIDCounter and the MessageID of the received GoodCRC message.		
	The Protocol Layer shall transition to the Message Sent state when: • The MessageIDCounter and the MessageID of the received GoodCRC message match.	6.6.1.1#1c	
	The Protocol Layer shall transition to the Check RetryCounter state when: • The MessageIDCounter and the MessageID of the received GoodCRC message do not match and • This is not a Soft Reset.	6.6.1.1#1d	
	The Protocol Layer shall transition to the Transmission Error state when: • The MessageIDCounter and the MessageID of the received GoodCRC message do not match and • This is a Soft Reset	6.3.12#2	
Subsection 6.8.2.1.6 MessageSent State			
	On entry to the Message Sent state the Protocol Layer shall increment the MessageIDCounter, reset the RetryCounter and inform the Policy Engine that the message has been sent.	6.6.1.1#1c	
	The Protocol Layer shall transition to the Wait for Message Request state when: • The Policy Engine has been informed that the message has been sent.		
Subsection 6.8.2.1.7 Check RetryCounter State			
	On entry to the Check RetryCounter state the Protocol Layer shall increment the value of the RetryCounter and then check it in order to determine whether it is necessary to retry sending the message.	6.6.1.1#1b	
	The Protocol Layer shall transition to the Construct Message state in order to retry message sending when: • RetryCounter ≤ nRetryCount.	6.6.1.1#1b	
	The Protocol Layer shall transition to the Transmission Error state when: • RetryCounter > nRetryCount.	6.6.1.1#1b	
Subsection 6.8.2.1.8 Transmission Error State			
	On entry to the Transmission Error state the Protocol Layer shall inform the Policy Engine of the transmission error.		
	The Protocol Layer shall transition to the Wait for Message Request state when: • The Policy Engine has been informed of the transmission error.		
Subsection 6.8.2.1.9 Discard Message State			
	Protocol Layer message transmission shall enter the Discard Message state whenever Protocol Layer Message reception receives an incoming message.		No other assertions for this.
	On entry to the the Discard Message state, if there is a message queued awaiting transmission, the Protocol Layer shall discard the message and increment the MessageIDCounter.		No other assertions for this.

Assertion	Description	Related Assertion	Comments
	The Protocol Layer shall transition to the Wait for Message Request state when: • Discarding is complete i.e. the message queue is empty.		No other assertions for this.
Subsection 6.8.2.2 Protocol Layer message Reception			
Subsection 6.8.2.2.1 Wait for First PHY Message State			
	This shall be the startup state for the Protocol Layer message reception and shall also be the state Protocol Layer message reception enters after completion of a Soft Reset or Hard Reset.		
	The Protocol Layer shall transition to the Initialize MessageID state when: • A received message is passed up from the PHY Layer • and the message is not a Soft Reset message.		
Subsection 6.8.2.2.2 Initialize MessageID State			
	On entry to the Initialize MessageID state the Protocol Layer shall transition Protocol Layer message transmission to the Discard Message state, store the MessageID of the incoming message and then pass the message to the Policy Engine.	6.6.1.2#1a	
	The Protocol Layer shall transition to the Send First GoodCRC state when: • The message has been passed to the Policy Engine.		
	The Protocol Layer shall transition to the Layer Reset for Receive state when: • A Soft Reset message is received from the PHY.		
Subsection 6.8.2.2.3 Send First GoodCRC State			
	On entry to the Send First GoodCRC state the Protocol Layer shall construct GoodCRC message and request the PHY to transmit it.		
	The Protocol Layer shall transition to the Wait for PHY Message state when: • The GoodCRC message has been passed to the PHY Layer.		
Subsection 6.8.2.2.4 Wait For PHY Message State			
	In the Wait For PHY Message state the Protocol Layer waits until the PHY Layer passes up a received message.		
	The Protocol Layer shall transition to the Send GoodCRC state when: • A message is passed up from the PHY Layer.		
	The Protocol Layer shall transition to the Layer Reset for Receive state when: • A Soft Reset message is received from the PHY.		
Subsection 6.8.2.2.5 Layer Reset for Receive State			
	On entry to the Layer Reset for Receive state the Protocol Layer shall reset the MessageIDCounter. The Protocol Layer shall transition Protocol Layer message transmission to the Wait Message Request state (see Section 6.8.2.1.1).	6.2.1.3#1	
	The Protocol Layer shall transition to the Send GoodCRC State when: • The Soft Reset actions in this state have been completed.		
Subsection 6.8.2.2.6 Send GoodCRC State			
	On entry to the Send GoodCRC state the Protocol Layer shall construct a GoodCRC message and request the PHY to transmit it.		

Assertion	Description	Related Assertion	Comments
	The Protocol Layer shall transition to the Check MessageID state when: • The GoodCRC message has been passed to the PHY Layer or • The PHY Layer indicates that a message has been discarded due to VBUS being busy but VBUS is now idle (see Section 5.8) • And a Soft Reset is NOT being performed.		
	The Protocol Layer shall transition to the Wait For First PHY Message state when: • The GoodCRC message has been passed to the PHY Layer or • The PHY Layer indicates that a message has been discarded due to VBUS being busy but VBUS is now idle (see Section 5.8) • And Soft Reset is being performed.		

10 Test Jig for Multiple Port Testing

When testing e.g. Hubs, there are some assertions which appear to require multiple Testers or a Tester with multiple ports. An alternative solution may be to define a simple 'device' which can plug into a second or further port of a hub, and provide sufficient functionality to allow the single tester to test the assertion in question on just one port.
The following is a list of tests which may be satisfied using this approach.

10.1 GotoMin Test

The Device would have PD Consumer capability, and would automatically request a given pre-settable voltage and current, and light an LED if successful.

10.2 Power Reserve Test

As above

10.3 MessageID per Port Test

The Device would have PD Consumer capability, and would engage the Hub in a series of message exchanges to change the transmit MessageID on that port. On removal, the Hub would send a Soft Reset and the Tester, which would be engaged in its own series of message exchanges, would check for the absense of a MessageID discontinuity.

10.3.1 Assertions Involved

6.3.2#2 The Source shall return power to the Sink(s) it has 'borrowed' from using the GotoMin mechanism before it can allocate any 'new' power to other devices.

> Needs a manufacturer specific mechanism to control the source to trigger a GotoMin and then wait for it to be returned after "x" time. What setup is required? Might require some sort of manual test setup to try this. Need to see the protocol messages.

6.6.1#2 Devices that support multiple ports, such as Hubs, shall maintain copies of the last MessageID on a per port basis.

> Requires to test on two ports, soft reset one when you are in the middle of a count and look for a change on the other.
> Simple plug in device?

11 System Policy Specific TDs

These tests are for PD Capable (PDUSB) devices and hubs.

All Chapter 9 tests follow the same initialization procedure. At the beginning of a test run, the host controller is reset and devices attached to the host are enumerated. If attached devices fail initialization assertions, the test tool will not be able to run the test. Note that the host controller is not reset in between tests if more than 1 test is selected.

If the DUT is a Hub, it must be run both as a peripheral (downstream of a PDUSB hub) and as the hub with a known good PD capable peripheral downstream.

11.1 Configuration Descriptor Test

Purpose	Verify Configuration Descriptor
Critical for Safety	No
Applies to	Devices and hubs
Description	Query DUT configuration descriptor and check pertinent PD fields.
Test setup	USB30CV; DUT must be connected to a PD capable port.
Preconditions	
Assertion References	9.1#1, 9.1#4
Checklist References	

11.1.1 Test Procedure

1. Enumerate and configure the DUT.
2. The DUT must be connected to a PD capable port.
3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    a. wValue set to 2 (Configuration)
    b. wIndex set to 0
    c. wLength set to 9d

    Test fails if the device does not return the descriptor or if the length returned is not 9d.

4. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    a. wValue set to 2 (Configuration)
    b. wIndex set to 0
    c. wLength set to wTotalLength (returned in previous call to GetDescriptor)
    Test fails if the device does not return the descriptor or if the length is not wTotalLength.

5. The DUT must report self-powered. Verify that Configuration.bmAttributes bit 6 is 1. (9.1#1).
6. The DUT must report bMaxPower field as 0. (9.1#4)

11.2 Power Delivery Capability Descriptor Test

Purpose	Verify Power Delivery Capability Descriptor
Critical for Safety	No
Applies to	Devices and hubs
Description	Verify Power Delivery Capability Descriptor.
Test setup	USB30CV; peripheral must be plugged into a PD capable port
Preconditions	
Assertion References	9.1.1#1, 9.2#1, 9.2#2, 9.2.1#1-9.2.1#18, 9.2.1#20-9.2.1#22
Checklist References	

11.2.1 Test Procedure

1. Enumerate and configure the DUT.
2. DUT must be connected to a PD capable port.
3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - wIndex set to 0
    - wLength set to 5d (All of the configuration Descriptor)

    Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1, 9.2#1, 9.2#2)

4. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - wIndex set to 0
    - wLength set to wTotalLength (returned in previous call to GetDescriptor)

5. Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1, 9.2#1, 9.2#2)

6. Check the size of the returned descriptor table. Test fails if the length of the returned descriptor table is not wTotalLength.
7. Parse the data, keeping the BOS Power Delivery Capability Descriptor. Perform the remaining steps on the BOS Power Delivery Capability Descriptor.
8. Test fails if there is not exactly 1 BOS Power Delivery Capability Descriptor contained in the BOS descriptor set. (9.2#2)
9. Test fails if bLength is not 16d. (9.2.3#17)

10. Test fails if bDescriptorType is not 10h. (9.2.1#18)
11. Test fails if bDevCapabilityType is not 6h. (9.2.1#19)
12. Test fails if bReserved is not 0. (9.2.1#1)
13. Test fails if bmAttributes bit 0 is not 0. (9.2.1.#2)
14. Prompt the user to find out if the device supports Battery Charging, and if so, what version and on which ports.
15. Compare user response to bmAttributes bit 1. A value of 1 indicates Battery Charging is supported. Test fails if bit 1 does not match user response. (9.2.1#3)
16. Compare user response to bcdBCVersion. Test fails if values do not match. (9.2.1#14)
17. Prompt the user to find out if the device supports Power Delivery, and if so, what version.
18. Compare user response to bmAttributes bit 2. A value of 1 indicates Power Delivery is supported. Test fails if bit 2 does not match user response. (9.2.1#4)
19. Compare user response to bcdPDVersion. Test fails if values do not match. (9.2.1#16)
20. Compare user response to bmAttributes bit 5. A value of 1 indicates that the peripheral supports the CHARGING_POLICY feature. The test fails if bit 5 does not match user response. (9.2.1#20)
21. Prompt the user to find out if the device supports USB Type-C, and if so, what version.
22. Compare user response to bmAttributes bit 6. A value of 1 indicates that the peripheral supports power capabilities defined in the USB Type C specification. The test fails if bit 6 does not match user response. (9.2.1#21)
23. Compare user response to bcdTypeCVersion. Test fails if values do not match. (9.2.1#22)
24. If device does not support Power Delivery, exit test.
25. Prompt the user to find out if the device is capable of being a power provider.
26. Compare user response to bmAttributes bit 3. A value of 1 indicates that peripheral can be a Power Provider. The test fails if bit 3 does not match user response. (9.2.1#5)
27. Prompt the user to find out if the peripheral is capable of being a power consumer.
28. Compare user response to bmAttributes bit 4. A value of 1 indicates that the peripheral can be a Power Consumer. The test fails if bit 4 does not match user response. (9.2.1#6)
29. Test fails if bits 7 of bmAttributes is not 0. (9.2.1#2)
30. Test fails if bit 8, 9 and 14 of bmAttributes are all 0. (9.2.1#7)
31. Test fails if bit 15 of bmAttributes is not 0. (9.2.1#2)
32. Test fails if bits 31:16 of bmAttributes are not 0. (9.2.1#2)
33. Prompt user to find out if the peripheral uses an AC power supply.
34. Compare user response to bit 8 of bmAttributes. A value of 1 indicates it supports AC power supply. If user response does not match, fail test. (9.2.1#8)
35. Prompt user to find out if the peripheral uses Battery power.
36. Compare user response to bit 9 of bmAttributes. A value of 1 indicates it supports a battery. If user response does not match, fail test. (9.2.1#9)
37. Prompt user to find out if the peripheral uses VBus.
38. Compare user response to bit 14 of bmAttributes. A value of 1 indicates it supports VBus. If user response does not match, fail test. (9.2.1#11)
39. Test fails if device is not a hub and any of the bits 1-15 of bmProviderPorts are non-zero. (9.2.1#12)
40. Test fails if Provider bit of bmAttributes is 1 and bmProviderPorts is 0. (9.2.1#12)
41. Test fails if Provider bit of bmAttributes is 0 and bmProviderPorts is non-zero. (9.2.1#12)

42. Test fails if peripheral is not a hub and bits 1:15 of bmProviderPorts are non-zero. (9.2.1#12)
43. Test fails if this is not a hub and any of the bits 1-15 of bmConsumerPorts are non-zero. (9.2.1#13)
44. Test fails if Consumer bit of bmAttributes is 1 and bmConsumerPorts is 0. (9.2.1#13)
45. Test fails if Consumer bit of bmAttributes is 0 and bmConsumerPorts is non-zero. (9.2.1#13)
46. Test fails if peripheral is not a hub and bits 1:15 of bmConsumerPorts are non-zero. (9.2.1#13)
47. If device is a hub, prompt the user to find out which ports are capable of being a power provider.
    - Test fails if bmProviderPorts does not match user response. (9.2.1#12)
48. If device is a hub, prompt the user to find out which ports are capable of being a power consumer.
    - Test fails if bmConsumerPorts does not match user response. (9.2.1#13)

11.3 Battery Info Capability Descriptor Test

Purpose	Verify Battery Info Capability Descriptor
Critical for Safety	No
Applies to	Devices and hubs
Description	Query UUT descriptors and check pertinent PD fields.
Test setup	USB30CV; peripheral must be plugged into a PD capable port
Preconditions	
Assertion References	9.1.1#1, 9.2#1, 9.2#2, 9.2.2#1, 9.2.2#2, 9.2.2#3, 9.2.2#4, 9.2.2#5, 9.2.2#7, 9.2.2#12-9.2.2#14
Checklist References	

11.3.1 Test Procedure

1. Enumerate and configure the DUT.
2. DUT must be connected to a PD capable port.
3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - wIndex set to 0
    - wLength set to 5d (All of the configuration Descriptor)

Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1 9.2#1, 9.2#2)

4. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)

- wIndex set to 0
- wLength set to wTotalLength (returned in previous call to GetDescriptor)

5. Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1 9.2#1, 9.2#2)
6. Check the size of the returned descriptor table. Test fails if the length of the returned descriptor table is not wTotalLength.
7. Parse the data, keeping the BOS Power Delivery Capability Descriptor and Battery Info Capability Descriptor(s).
8. Test fails if bLength is not 24d. (9.2.2#12)
9. Test fails if bDescriptorType is not 10h. (9.2.2#13)
10. Test fails if bDevCapabilityType is not 7h. (9.2.2#14)
11. If bmPowerSource field bit 9 (Battery) in the PD Capability Descriptor is 0, fail the test if the device returned a Battery Info Capability Descriptor. (9.2.2#1)
12. If bmPowerSource field bit 9 (Battery) in the PD Capability Descriptor is 1, fail the test if the device did not return a Battery Info Capability Descriptor (9.2.2#1)
13. Test fails if the number of Battery Info Capability Descriptors does not match bmPowerSource field bits 11-13 (NumBatteries) in PD Capability Descriptor. (9.2.2#2)
14. Validate the string descriptor referenced by iBattery. (9.2.2#3)
15. Print the iBattery string descriptor.
16. Validate the string descriptor referenced by iSerial is present. (9.2.2#4)
17. Print the iSerial string descriptor.
18. Validate the string descriptor referenced by iManufacturer. (9.2.2#5)
19. Print the iManufacturer string descriptor.
20. Print bBatteryID value.
21. Test fails if bReserved is not 0. (9.2.2.#7)
22. Print dwChargedThreshold, dwWeakThreshold, dwBatteryDesignCapacity and dwBatterylastFullChargeCapacity.

11.4 PD Consumer Port Capability Descriptor Test

Purpose	Verify Consumer Port Capability Descriptors
Critical for Safety	No
Applies to	Devices and hubs
Description	Query DUT descriptors and check pertinent PD fields.
Test setup	USB30CV; peripheral must be plugged into a PD capable port
Preconditions	
Assertion References	9.1.1#1, 9.2#1, 9.2#2, 9.2.3#1, 9.2.3#2, 9.2.3#4- 9.2.3#7, 9.2.3#15-9.2.3#18
Checklist References	

11.4.1 Test Procedure

1. Enumerate and configure the DUT.
2. DUT must be connected to a PD capable port.
3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - wIndex set to 0
    - wLength set to 5d (All of the configuration Descriptor)

Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1, 9.2#1, 9.2#2)

4. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    - wValue set to 15d (BOS)
    - wIndex set to 0
    - wLength set to wTotalLength (returned in previous call to GetDescriptor)

Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1, 9.2#1, 9.2#2)

5. Parse the data, keeping the BOS Power Delivery Capability Descriptor and Power Delivery Consumer Port Capability Descriptor(s).
6. Test fails if the number of PD Consumer Port Capability Descriptors does not match bmConsumerPorts field in PD Capability Descriptor. (9.2.3#1, 9.2.3#2)
7. Test fails if there is more than one PD Consumer Port Capability Descriptor for each port that is capable of consuming power. (9.2.3#2)
8. Test fails if the DUT is not a hub and it has more than 1 consumer capable port. (9.2.3#4)
9. Test fails if the DUT is a hub and it has more than 16 consumer capable ports. (9.2.3#5)
   For each Consumer Port Capability Descriptor:
10. Test fails if bLength is not set to 24d. (9.2.3#15)
11. Test fails if bDescriptorType is not set to 10h. (9.2.3#16)
12. Test fails if bDevCapability is not set to 8h. (9.2.3#17)
13. Test fails if bits 15:3 of bmCapabilities is not set to 0. (9.2.3#7)
14. Test fails if bReserved field is not set to 0. (9.2.3#6)
15. Test fails if wReserved field is not set to 0. (9.2.3#10)
16. Print bmCapabilities, wMinVoltage, wMaxVoltage and dwMaxPeakPowerTime.

11.5 PD Provider Port Capability Descriptor Test

Purpose	Verify Provider Port Capability Descriptor
Critical for Safety	No
Applies to	Devices and hubs
Description	Query DUT descriptors and check pertinent PD fields.

Test setup	USB30CV; peripheral must be plugged into a PD capable port
Preconditions	
Assertion References	9.1.1#1, 9.2#1, 9.2#29.2.4#1, 9.2.4#2, 9.2.4#4, 9.2.4#5, 9.2.4#6, 9.2.4#8, 9.2.4#12-9.2.4#16
Checklist References	

11.5.1 Test Procedure

1. Enumerate and configure the DUT.
2. DUT must be connected to a PD capable port.
3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
   - wValue set to 15d (BOS)
   - wIndex set to 0
   - wLength set to 5d (All of the configuration Descriptor)

Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1, 9.2#1, 9.2#2**)**

4. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
   - wValue set to 15d (BOS)
   - wIndex set to 0
   - wLength set to wTotalLength (returned in previous call to GetDescriptor)

Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1, 9.2#1, 9.2#2)

5. Parse the data, keeping the BOS Power Delivery Capability Descriptor and Power Delivery Provider Port Capability Descriptor(s).
6. Test fails if the number of PD Provider Port Capability Descriptors does not match bmProviderPorts field in PD Capability Descriptor. (9.2.4#1, 9.2.4#2)
7. Test fails if the DUT is not a hub and has more than 1 provider capable port. (9.2.3#4)
8. Test fails if the DUT is a hub and it has more than 16 provider capable ports. (9.2.4#5)
   For each Provider Port Capability Descriptor:
9. Test fails if bLength is not 8d + (4d * bNumOfPDObjects). (9.2.4#8, 9.2.4#12)
10. Test fails if bDescriptorType is not 10h. (9.2.4#12)
11. Test fails if bDevCapabilityType is not 9h. (9.2.4#13)
12. Test fails if bReserved fields at offset 3 and 7 are not set to 0. (9.2.4#6)
13. Print bmCapabilities values.
14. Test fails if bReserved field is not 0. (9.2.4#16)
15. Test fails if bits 15:3 of bmCapabilities field is not set to 0. (9.2.4#15)
16. Print bNumOfPDObjects value.
17. Print all PowerDataObjects.

11.6 Battery Status Test

Purpose	Verify Battery Status reporting
Critical for Safety	No
Applies to	Peripherals and hubs
Description	
Test setup	USB30CV
Preconditions	
Assertion References	9.4.2#1 – 9.4.2#15, 9.2.1#12, 9.2.2#2, 9.4.2#21-9.4.2#23
Checklist References	

11.6.1 Test Procedure

1. Prompt the user to connect the DUT with all batteries attached and, if present, AC supply attached.
2. Enumerate and configure the DUT behind a PDUSB Hub.
3. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    a. wValue set to 15d (BOS)
    b. wIndex set to 0
    c. wLength set to 5d (All of the configuration Descriptor)
4. Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1, 9.2#1, 9.2#2)
5. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
    a. wValue set to 15d (BOS)
    b. wIndex set to 0
    c. wLength set to wTotalLength (returned in previous call to GetDescriptor)
6. Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1, 9.2#1, 9.2#2)
7. Parse the data, keeping the BOS Power Delivery Capability Descriptor and Battery Info Capability Descriptor(s).
8. If bit 9 of USB Power Delivery Capability Descriptor is 0, device does not have a battery. Steps for device that does not have a battery:
9. If device returned any Battery Info Capability Descriptors, fail test. (9.2.2#1)
10. Skip remaining steps in this test.
11. Prompt user for the number of batteries present.
12. Verify that the number of batteries reported by the user matches Battery info Capability Descriptor bmPowerSource NumBatteries (bits 11:13). (9.2.1#10,9.2.2#2)
13. Verify that the number of batteries reported by the user matches the number of Battery Info Descriptors returned by the DUT. (9.2.1#10,9.2.2#2)

Test Case for Attached Battery:

14. For each Battery Info Descriptor:
    a. Issue GetBatteryStatus request
        i. bRequest set to 21 (GET_BATTERY_STATUS)
        ii. wValue set to 0
        iii. wIndex set to bBatteryID from Battery Info Descriptor
        iv. wLength set to 8
    b. Test fails if device does not return 8 bytes. (9.4.2#1, 9.4.2#3, 9.4.2#4)
    c. Test fails if byte 0 of returned data is 0, indicating no battery present. (9.4.2#5)
    d. Prompt user for charging/discharging state of this battery. Note that if the peripheral has external power plugged in, the battery is charging. If the peripheral does not have external power or the external power is unplugged, the battery is discharging.
    e. If battery is charging, byte 0 of returned data must be 1. (9.4.2#21)
    f. If battery is discharging, byte 0 of returned data must be 2. (9.4.2#22)
    g. If battery is neither discharging nor charging, byte 0 of returned data must be 3. (9.4.2#23)
    h. Test fails if byte 0 returns a value greater than 3. (9.4.2#6)
    i. Print out Battery SOC (byte 1), Remaining Operating time (bytes 4 and 5) and Remaining Charge time (bytes 6 and 7).
    j. Test fails if byte 2 (bBatteryStatus) of returned data is greater than 7. (9.4.2#9)
    k. Test fails if byte 3 (bRemoteWakeCapStatus) of returned data is non-zero. (9.4.2#12, 9.4.2#13)

Test Case for Detached Battery:

15. For all Battery IDs:
    a. Prompt user to remove battery associated with Battery ID being tested.
    b. Re-enumerate and configure DUT.
        i. If re-enumeration fails, then DUT cannot function without the battery attached. Skip the remainder of this step.
    c. Issue GetBatteryStatus request
        i. bRequest set to 21 (GET_BATTERY_STATUS)
        ii. wValue set to 0
        iii. wIndex set to bBattery ID from Battery Info Descriptor
        iv. wLength set to 8
    d. Test fails if device does not return 8 bytes. (9.4.2#1, 9.4.2#3, 9.4.2#4)
    e. Test fails if byte 0 of returned data is not 0. (9.4.2#5)
    f. Print out Battery SOC (byte 1), Remaining Operating time (bytes 4 and 5) and Remaining Charge time (bytes 6 and 7).
    g. Test fails if byte 3 (bRemoteWakeCapStatus) of returned data is non-zero. (9.4.2#12, 9.4.2#13)
16. Prompt user to re-attach battery.

11.7 Remote Wake Test

Purpose	Verify that appropriate battery events generate a USB wake.
Critical for Safety	No
Applies to	All PD devices that have batteries
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.2#11, 9.4.5#1
Checklist References	

11.7.1 Test Procedure

1. Enumerate and configure the DUT.
2. Determine if DUT has batteries:
    a. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
        i. wValue set to 15d (BOS)
        ii. wIndex set to 0
        iii. wLength set to 5d (All of the configuration Descriptor)
        iv. Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1, 9.1.1#3)

    b. Send a GetDescriptor(wValue, wIndex, wLength, Data) request with the following values:
        i. wValue set to 15d (BOS)
        ii. wIndex set to 0
        iii. wLength set to wTotalLength (returned in previous call to GetDescriptor)
        iv. Test fails if the device does not return the descriptor or if the length is not wTotalLength. (9.1.1#1, 9.1.1#3)

    c. Check the size of the returned descriptor table. Test fails if the length of the returned descriptor table is not wTotalLength.
    d. Parse the data, keeping the BOS Power Delivery Capability Descriptor.
3. If bit 9 of bmPowerSource of USB Power Delivery Capability Descriptor is 0, device does not have batteries. End test.
4. Determine if the DUT is remote wake capable:
    a. If the device is LS, FS or HS:
        i. Read the configuration descriptor for the DUT.
        ii. If bit 5 of the bmAttributes field is set, this device supports Remote Wake.
    b. If the device is SuperSpeed (Gen x)
        i. Read the full configuration descriptor for the DUT.
        ii. For each interface contained in the configuration descriptor:

1. Send GetStatus request to the interface.

For each battery:

5. If bit 1 of the data returned from GetStatus is 1, this device supports Remote Wake.
6. Prompt user to remove battery.
7. Attempt to re-enumerate and configure device.
   a. If there is no device detected, then DUT requires battery in order for host to detect it.
   b. Prompt user to reinsert battery.
   c. Re-enumerate and configure device.
   d. Go on to next battery.
8. Issue Set PD Feature to DUT:
   a. bRequest set to SET_FEATURE
   b. wValue set to 40BATTERY_WAKE_MASK)
   c. wIndex set to 1 (Wake on Battery Present bit 0
9. If DUT is a hub, then issue SetPortFeature (PORT_SUSPEND) on all downstream ports.
10. Suspend the device.
    a. If device is LS/FS/HS:
       i. Issue SetPortFeature (PORT_SUSPEND) to Hub downstream port where DUT is attached.
       ii. Test fails if SetPortFeature call fails.
    b. If device is SS:
       i. Issue SetFeature (FUNCTION_SUSPEND) to all interfaces that support function suspend.
       ii. Test fails any SetPortFeature call fails.
11. Issue SetPortFeature (C_REMOTE_WAKE_MASK) on upstream port partner to enable remote wake on downstream port (host or hub).
12. Test fails if SetFeature call fails.
13. Issue SetPortFeature (PORT_SUSPEND) on downstream port.
14. Test fails if SetPortFeature call fails.
15. Issue GetPortStatus and wait until the port is suspended.
16. Prompt user to insert the DUT battery. If DUT does not support remote wake:
17. Test fails if DUT resumed from suspend. (assert #) If DUT does support remote wake:
18. Make sure link wakes <expand this>
19. Make sure remote wake event is there for SS <expand this>
20. Read PD stuff and make sure it is correct. <expand this> (9.4.5#1)

11.8 Hub Get Port Partner PDO Test

Purpose	Verify that the PDO list returned via USB descriptor is the same PDO list returned via PD mechanisms.

Critical for Safety	No
Applies to	Hubs and peripherals
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.3#3, 9.4.3#5
Checklist References	

11.8.1 Test Procedure

1. This test can be run on either a PDUSB hub, or a PD-capable device connected on a downstream facing port of a PDUSB hub. If it is being run on a hub, then it must be run on all PD-capable downstream facing ports.
2. Enumerate and configure the DUT.
3. If the test is being run on a hub:
    a. prompt the user to attach a device in which the upstream facing port can be a Provider.
    b. Enumerate and configure the device.
4. Send a GetDescriptor(wValue, wIndex, wLength, Data) request to the downstream device with the following values:
    a. wValue set to 15d (BOS)
    b. wIndex set to 0
    c. wLength set to 5d (All of the configuration Descriptor)
        i. Test fails if the device does not return the descriptor or if the length returned is not 5d. (9.1.1#1, 9.1.1#3**)**
5. Send a GetDescriptor(wValue, wIndex, wLength, Data) request to the downstream device with the following values:
    a. wValue set to 15d (BOS)
    b. wIndex set to 0
    c. wLength set to wTotalLength (returned in previous call to GetDescriptor)
    d. Check the size of the returned descriptor table. Test fails if the length of the returned descriptor table is not wTotalLength. (9.1.1#1, 9.1.1#3)
    e. Parse the data, saving Power Delivery Provider Port Capability Descriptor(s).
6. If bNumOfPDObjects field of PD Provider Port Capability Descriptor is 0, skip Provider PDO verification.
7. Send Get Partner PDO request to hub.
    a. wValue set to 20d (GET_PARTNER_PDO)

  b. wValue set to 1 (provider)
  c. wIndex set to port number that DUT is connected to.
  d. wLength set to 2 + bNumOfPDOObjects * 4. (header + num of PDOs expected)
  e. Test fails if request fails or is stalled by the DUT. (9.4.3#3)
  f. Test fails if the request returns less than (2 + bNumOfPDOObjects * 4) bytes of data. (9.4.3#3)
8. Compare the header and PDO list returned through GetBOSDescriptor () and GET_PARTNER_PDO request. Test fails if the data is different. (9.4.3#3).
9. < todo make sure all capabilities work with a compliance device>

Repetitions:

If testing the PDUSB hub, repeat this test for all hub downstream ports.

11.9 External Supply Test

Purpose	Verify reporting of External Supply of DUT
Critical for Safety	No
Applies to	PD capable peripheral and PDUSB hub
Description	Remove supply of DUT connected DS of hub. Device must report external supply change, hub must report this also in PD Status bits.
Test setup	USB30CV; PD capable peripheral
Preconditions	DUT must have external supply.
Assertion References	9.4.4#5, 9.4.4#6, 9.4.4#7
Checklist References	

11.9.1 Test Procedure

1. This test can be run on either a PDUSB hub, or a PD-capable device connected on a downstream facing port of a PDUSB hub. If it is being run on a hub, then it must be run on all PD-capable downstream facing ports.
2. Enumerate and configure DUT.
3. If the test is being run on a hub:
   a. prompt the user to attach a device with the following properties:
      i. it has an external power supply.
      ii. it can function when the external power supply is removed.
   b. Enumerate and configure the device.
4. Clear any change bits in hub, including any Port PD change bits.
5. If downstream device does not support an external power supply, then exit the test.
6. If downstream device supports an external power supply do the following steps:
   a. Read PD Port Status by sending GetStatus request with the following values:

i. wValue set to 1.
ii. wIndex set to port number.
iii. wLength set to 8d.

b. Test fails if GetStatus fails or does not return 8 bytes (9.4.4#1)
c. Verify that all reserved bits of returned data are 0. (9.4.4#3)
d. Verify that bit 16 of returned data is 1, indicating externally powered. (9.4.4#22)

7. Prompt user to remove external power supply on downstream device.

8. If device can operate without external supply do the following steps:
   a. Read PD Port Status by sending GetStatus request with the following values:
      i. wValue set to 1.
      ii. wIndex set to port number.
      iii. wLength set to 8d.
   b. Test fails if GetStatus fails or does not return 8 bytes (9.4.4#1)
   c. Verify that all reserved bits of returned data are 0.
      (9.4.4#3) If this is the port the DUT is attached to:
   d. Verify that bit 1 (External Supply Change) of returned data is 1, indicating change in the external supply (9.4.4#4)
   e. Verify that bit 16 (Supply) is set to 0 (No Supply)
      (9.4.4#22) If this is not the DUT port:
   f. Verify that bit 1 (External Supply Change) of returned data is 0, indicating no change.
9. Clear change bits
10. Prompt user to add external supply to downstream device.
   a. Read PD Port Status by sending GetStatus request with the following values:
      i. wValue set to 1.
      ii. wIndex set to port number.
      iii. wLength set to 8d.
   b. Test fails if GetStatus fails or does not return 8 bytes (9.4.4#1)
   c. Verify that all reserved bits of returned data are 0.
      (9.4.4#3) If this is the port the DUT is attached to:
   d. Verify that bit 1 (External Supply Change) of returned data is 1, indicating change in the external supply (9.4.4#4)
   e. Verify that bit 16 (Supply) is set to 1 (Supply)
      (9.4.4#22) If this is not the DUT port:
   f. Verify that bit 1 (External Supply Change) of returned data is 0, indicating no change.

11.10 Hub Cable Test

Purpose	Verify that hub correctly reports attached cable status.
Critical for Safety	No
Applies to	Hubs

Description	
Test setup	USB30CV; legacy USB cable and PDUSB cable
Preconditions	
Assertion References	9.4.4#5, 9.4.4#7, 9.4.4#21, 9.4.4#29, 9.4.4#30
Checklist References	

11.10.1 Test Procedure

1. Enumerate and configure a PDUSB hub with no downstream devices attached.
2. If port under test does not support PD, then exit the test.

No Cable Attached

3. Read Port Status by sending GetStatus request with the following values:
   a. wValue set to 0 (PORT_STATUS).
   b. wIndex set to port number.
   c. wLength set to 4d.
4. Test fails if GetStatus fails or does not return 4 bytes (assert #)
5. For all ports, check these bits of the port status:
   a. C_PORT_CONNECTION
   b. C_PORT_OVER_CURRENT
   c. C_RESET_CHANGE
   d. C_PORT_ENABLE (if a USB 2 hub)
   e. C_PORT_SUSPEND (if a USB 2 hub)
   f. C_BH_PORT_RESET (if a SS hub)
   g. C_PORT_LINK_STATE (if a SS hub)
   h. C_PORT_CONFIG_ERROR (if a SS hub)
   i. Bit 15 (C_PORT _PD_CHANGE)
6. If any of these bits other than C_PORT_PD_CHANGE are set, issue ClearPortFeature request for the appropriate change bit.
7. If C_PORT_PD_CHANGE is set then clear the PD Port Status change bits:
   a. Read PD Port Status for that port with a GetPortStatus () request.
   b. If bits 0:15 of the PD Port Status are not 0, then clear the PD status bits by sending ClearPortPDFeature for Port Under Test with bits 0:15 of Port PD Status.
8. Read PD Port Status by sending GetStatus request with the following values:
   a. wValue set to 1 (PD_STATUS).
   b. wIndex set to port number.
   c. wLength set to 8d.
9. Test fails if GetStatus fails or does not return 8 bytes (9.3.1#1, 9.3.1#2, 9.4.4#1)
10. Check reserved bits:
    a. Test fails if bit 0 of Port PD Status is non-zero. (9.4.4#3)

   b. Test fails if bit 10 of Port PD Status is non-zero. (9.4.4#3)
   c. Test fails if bit 14 of Port PD Status is non-zero. (9.4.4#3)
11. Test fails if bit 20 (Insertion Detect) is non-zero. (9.4.4#25)
12. Test fails if bit 21 (PD Capable Cable) is non-zero. (9.4.4#26)

Legacy Cable Attach

13. Prompt user to attach a legacy cable (no peripheral) to a port.
14. Send read request of 4 bytes to hub status change endpoint.
15. If request fails, test fails (assert)
16. Verify that exactly 1 bit in the port change status is set to 1. (assert) The set bit indicates the Port Under Test (PUT).
17. Read PD Port Status by sending GetStatus request with the following values:
    a. wValue set to 1.
    b. wIndex set to PUT.
    c. wLength set to 8d.
18. Test fails if GetStatus fails or does not return 8 bytes (9.3.1#1, 9.3.1#2, 9.4.4#1)
19. Check reserved bits:
    a. Test fails if bit 0 of Port PD Status is non-zero. (9.4.4#3)
    b. Test fails if bit 10 of Port PD Status is non-zero. (9.4.4#3)
    c. Test fails if bits 13:14 of Port PD Status is non-zero. (9.4.4#3)
20. Bit 20 (Insertion Detect) must be one for PUT. (9.4.4#25)
21. Bit 20 (Insertion Detect) but be zero for other ports. (9.4.4#25)
22. Bit 21 (PD Capable Cable) must be zero for all ports. (9.4.4#26)
23. Prompt user to detach cable.
    a. Read PD Port Status for that port with a GetPortStatus () request.
    b. If bits 0:15 of the PD Port Status are not 0, then clear the PD status bits by sending ClearPortPDFeature for Port Under Test with bits 0:15 of Port PD Status.
24. Test fails if ClearPortFeature () fails. (9.3.1#1, 9.3.1#2)

PD Capable Cable

25. Prompt user to attach a Power Delivery capable cable (only) to a port.
26. Verify that exactly 1 bit in the port change status is set to 1. (assert) The set bit indicates the Port Under Test (PUT).
27. Read PD Port Status by sending GetStatus request with the following values:
    a. wValue set to 1.
    b. wIndex set to PUT.
    c. wLength set to 8d.
28. Test fails if GetStatus fails or does not return 8 bytes (9.3.1#1, 9.3.1#2, 9.4.4#1)
29. Check reserved bits:

   a. Test fails if bit 0 of Port PD Status is non-zero. (9.4.4#3)
   b. Test fails if bit 10 of Port PD Status is non-zero. (9.4.4#3)
   c. Test fails if bits 13:14 of Port PD Status is non-zero. (9.4.4#3)
30. Bit 20 (Insertion Detect) must be one for PUT. (9.4.4#25)
31. Bit 20 (Insertion Detect) must be zero for other ports. (9.4.4#25)
32. Bit 21 (PD Capable Cable) must be one for PUT. (9.4.4#26)
33. Bit 21 (PD Capable Cable) must be zero for other ports. (9.4.4#26)
34. Prompt user to detach cable.

11.11 Hub Set Port PDO Test

Critical for Safety	No
Applies to	Hubs
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.7#1, 9.4.7#2, 9.4.7#10, 9.4.7#4, 9.4.7#11
Checklist References	

11.11.1 Test Procedure

1. Enumerate and configure the HUT.
2. If port under test does not support provider capabilities, then exit the test.
3. How do we get software to make it active as a provider? Does it just depend on the device connected?
4. Enable PD notifications by sending SetFeature request with the following values:
   a. wValue set to 41d (OS_IS_PD_AWARE).
   b. wIndex set to 1 (SPM is active on host).
   c. wLength set to 0.
   d. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
5. Set hub to Intrusive mode by sending SetFeature request with the following values:
   a. wValue set to 42d (POLICY_MODE).
   b. wIndex set to 1 (Intrusive mode).
   c. wLength set to 0.
   d. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Read the PD Provider Port Capability Descriptors for the hub.
7. For each PDO in the list:

   a. Send SetPortPDOs request with 1 PDO.
   b. Verify that attached device received updated capabilities. [tester/compliance device/analyzer] (9.4.7#1)
8. Send all PDOs at one time
   a. Send SetPortPDOsrequest with all PDOs.
   b. Verify that attached device received updated capabilities. [tester/compliance device/analyzer]
9. Send invalid PDO list
   a. Add 1 more PDO to the list read from PD Provider Port Capability Descriptor.
   b. Send SetPortPDOs request with this invalid list of PDOs.
   c. Hub must respond with a request error (9.4.7#3)
10. Return hub to non-intrusive mode by sending SetFeature request with the following values:
11. wValue set to 41d (OS_IS_PD_AWARE).
12. wIndex set to 0 (SPM is not active on host).
13. wLength set to 0.
14. Test fails if SetFeature () fails.
15. Repeat for each downstream facing port.

11.12 Hub Change Port Operation Test

Purpose	Verify hub ability to change port operation.
Critical for Safety	No
Applies to	Hubs
Description	
Test setup	USB30CV; legacy peripheral, Battery Charging capable peripheral that does not support PD, PD capable peripheral
Preconditions	
Assertion References	9.4.4#1, 9.4.4#5, 9.4.4#24
Checklist References	

11.12.1 Test Procedure

1. Enumerate and configure a PDUSB hub with no devices attached.
2. For all downstream ports
   a. Read Port Status request by sending GetStatus request with the following values:
      i. wValue set to 0.
      ii. wIndex set to port number.
      iii. wLength set to 4d.
      iv. Test fails if GetPortStatus fails or does not return 4 bytes.

b. If the C_PORT_PD_CHANGE bit is set, then Read PD Port Status by sending Get Status request with the following values:
   i. wValue set to 1.
   ii. wIndex set to port number.
   iii. wLength set to 8d.
   iv. Test fails if GetStatus fails or does not return 8 bytes (9.4.4#1).
c. If bits 0:15 are not 0, then clear the PD status bits by sending ClearPortPDFeature for Port Under Test with bits 0:15 of Port PD Status.

3. For each port, determine whether it supports PD (that is, it's is a PD Provider or PD Consumer or both).
   a. If it does not support PD, then exit the test.
4. Prompt user to attach a Legacy USB device.
5. Read PD Port Status by sending GetStatus request with the following values:
   a. wValue set to 1.
   b. wIndex set to port number of attached device.
   c. wLength set to 8d.
   d. Test fails if GetStatus fails or does not return 8 bytes (9.4.4#1)
6. Verify that bits 17:19 of returned data are set to the value 1, indicating the port is in legacy mode. (9.4.4#24)
7. Verify that Port Operation Mode bit (2) is set. (9.4.4#5)
8. Prompt user to detach device.
9. If bits 0:15 are not 0, then clear the PD status bits by sending ClearPortPDFeature for Port Under Test with bits 0:15 of Port PD Status.
10. Determine whether the current port supports Battery Charging (see the Power Delivery Capability Descriptor, and the Port Consumer Descriptor and Port Provider Descriptor for that port)
    a. If the port does not support Battery Charging, skip to step 18.
11. Prompt user to attach a Battery Charging capable device that does not support PD.
12. Read PD Port Status by sending GetStatus request with the following values:
    a. wValue set to 1.
    b. wIndex set to port number of attached device.
    c. wLength set to 8d.
    d. Test fails if GetStatus fails or does not return 8 bytes (9.4.4#1)
13. Verify that bits 17:19 of returned data set to the value 2, indicating the port is in Battery Charging mode. (9.4.4#24)
14. Verify that Port Operation Mode bit (2) is set. (9.4.4#5)
15. Prompt user to detach device.
16. If bits 0:15 are not 0, then clear the PD status bits by sending ClearPortPDFeature for Port Under Test with bits 0:15 of Port PD Status.
17. Prompt user to attach a Power Delivery capable device.
18. Read PD Port Status by sending GetStatus request with the following values:
    a. wValue set to 1.
    b. wIndex set to port number of attached device.
    c. wLength set to 8d.
    d. Test fails if GetStatus fails or does not return 8 bytes (9.4.4#1)

19. Verify that Port Operation Mode bit (2) is set. (9.4.4#5)
20. Verify that bits 17:19 of returned data is set to 3, indicating the port is in Power Delivery mode. (9.4.4#24)
21. If bits 0:15 are not 0, then clear the PD status bits by sending ClearPortPDFeature for Port Under Test with bits 0:15 of Port PD Status.
22. Detach peripheral.
23. Read PD Port Status by sending GetStatus request with the following values:
    a. wValue set to 1.
    b. wIndex set to port number.
    c. wLength set to 8d.
    d. Test fails if GetStatus fails or does not return 8 bytes (9.4.4#1)
24. Verify that bits 17:19 of returned data are set to 0, indicating the port is in No Consumer mode. (9.4.4#24)
25. Verify that Port Operation Mode bit (2) is set. (9.4.4#5)
26. Repeat test for all downstream facing ports.

11.13 Hub Intrusive Mode SWAP Test

Purpose	Verify SWAP in intrusive mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11, 9.4.6#12, 9.3.2#1, 9.3.2#2, 9.3.2#3. 9.4.4#12
Checklist References	

11.13.1 Test Procedure

1. Enumerate and configure the DUT behind PDUSB Hub Under Test.
2. Notify hub that OS is PD Aware by sending SetFeature() request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Intrusive mode by sending SetFeature() request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 1 (Intrusive mode).
    c. wLength set to 0.

5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
    e. Save the value of the PD Direction bit (22).
7. Initiate a SWAP by sending SetFeature with the following values:
    a. wValue set to 43d (SWAP).
    b. wIndex set to port number of attached device.
    c. wLength set to 0.
8. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
9. Issue read to interrupt IN endpoint of hub.
10. Test fails if read fails (9.3.2#1)
11. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
12. Test fails if any other ports show status changes.
13. Read Port Status and Change bits for port under test.
14. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
15. Test fails if any other Port Status change bits are set.
16. Read Port Status for all ports not under test. Verify that no change bits are set.
17. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
18. Test fails if New Power Direction bit (5) is set. (9.4.4#9)
19. Test fails if Swap Complete bit (12) is not set. (9.4.4#20)
20. Test fails if PD Direction bit (22) is not opposite of what it was before SWAP.(9.4.4#20)
21. Send ClearPortPDFeature for Port Under Test with Swap Complete bit (12) set.
22. Read PD Port Status for port under test.
23. Test fails if Swap Complete bit (12) is not 0. (9.4.1#4)
24. Read PD Port Status for ports not under test. Verify that no change bits are set.
25. Repeat steps to swap the direction back to the original orientation.
26. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
27. Test fails if SetFeature () request fails.
28. Repeat test for all downstream facing ports.

11.14 Hub Intrusive Mode GOTO_MIN Test

Purpose	Verify GOTO_MIN in intrusive mode

Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11
Checklist References	

11.14.1 Test Procedure

1. Enumerate and configure PD-capable peripheral behind PDUSB Hub Under Test.
2. Enable PD notifications by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
3. wLength set to 0.
4. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
5. Set hub to Intrusive mode by sending SetFeature() request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 1 (Intrusive mode).
    c. wLength set to 0.
6. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
7. Make sure the hub DS port is acting as a Provider.
8. Initiate a GOTO_MIN by sending SetFeature with the following values:
    a. wValue set to 44d (GOTO_MIN).
    b. wIndex set to port number of attached device.
    c. wLength set to 0.
9. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
10. Verify that the GOTO_MIN request was sent to the port partner. [tester/compliance device can verify this was received]
11. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
12. Repeat test for all downstream facing ports.

11.15 Hub Intrusive Mode RETURN_POWER Test

Purpose	Verify RETURN_POWER in intrusive mode
Critical for Safety	No

Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11
Checklist References	

11.15.1 Test Procedure

1. Enumerate and configure the PD-capable peripheral behind PDUSB Hub Under Test.
2. Enable PD notifications by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 1 (Intrusive mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Initiate a RETURN_POWER by sending SetFeature with the following values:
    a. wValue set to 45d (RETURN_POWER).
    b. wIndex set to port number of attached device.
    c. wLength set to 0.
7. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
8. Verify that the hub returned power to the port partner. [tester/compliance device can verify this was received] (9.4.6#8)
9. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
10. Repeat test for all downstream facing ports.

11.16 Hub Intrusive Mode Negotiate Power Level ACCEPT_PD_REQUEST Test

Purpose	Verify ACCEPT_PD_REQUEST in intrusive mode
Critical for Safety	No

Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11
Checklist References	

11.16.1 Test Procedure

1. Enumerate and configure the PD-capable peripheral behind PDUSB Hub Under Test.
2. Enable PD notifications by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 1 (Intrusive mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Have DUT initiate a request to renegotiate power [tester/compliance device]

Make sure hub reports this request:

7. Issue read to interrupt IN endpoint of hub.
8. Test fails if read fails (9.3.2#1)
9. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
10. Test fails if any other ports show status changes.
11. Read Port Status and Change bits for port under test.
12. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
13. Test fails if any other Port Status change bits are set.
14. Read Port Status for all ports not under test. Verify that no change bits are set.
15. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
16. Test fails if Negotiated Power Level bit (4) is not set. (9.4.4#8)
17. Verify that the power level specified in Request Data Object is the correct power level.
18. Send ClearPortPDFeature for Port Under Test with Negotiated Power Level (4) set.
19. Read PD Port Status for port under test.
20. Verify Negotiated Power Level bit has been cleared.

Accept request:

21. Initiate a ACCEPT_PD_REQUEST by sending SetFeature with the following values:
    a. wValue set to 46d (ACCEPT_PD_REQUEST).
    b. wIndex set to port number of attached device.
    c. wLength set to 0.
22. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
23. Verify that power was renegotiated [tester/compliance device/analyzer]
24. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
25. Repeat test for all downstream facing ports.

11.17 Hub Intrusive Mode Swap ACCEPT_PD_REQUEST Test

Purpose	Verify ACCEPT_PD_REQUEST in intrusive mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11
Checklist References	

11.17.1 Test Procedure

1. Enumerate and configure the PD-capable peripheral behind PDUSB Hub Under Test.
2. Enable PD notifications by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 1 (Intrusive mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.

   c. wLength set to 8d.
   d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
   e. Save the value of the PD Direction bit (22).
7. Have DUT initiate a SWAP command[tester/compliance device]

Make sure hub reports this request:

8. Issue read to interrupt IN endpoint of hub.
9. Test fails if read fails (9.3.2#1)
10. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
11. Test fails if any other ports show status changes.
12. Read Port Status and Change bits for port under test.
13. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
14. Test fails if any other Port Status change bits are set.
15. Read Port Status for all ports not under test. Verify that no change bits are set.
16. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
17. Test fails if New Power Direction bit (5) is not set. (9.4.4#12)
18. Send ClearPortPDFeature for Port Under Test with New Power Direction bit (5) set.
19. Read PD Port Status for port under test.
20. Verify New Power Direction bit has been cleared.

Accept request:

21. Initiate a ACCEPT_PD_REQUEST by sending SetFeature with the following values:

22. wValue set to 46d (ACCEPT_PD_REQUEST).
23. wIndex set to port number of attached device.
24. wLength set to 0.
25. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
26. Issue read to interrupt IN endpoint of hub.
27. Test fails if read fails (9.3.2#1)
28. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
29. Test fails if any other ports show status changes.
30. Read Port Status and Change bits for port under test.
31. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
32. Test fails if any other Port Status change bits are set.
33. Read Port Status for all ports not under test. Verify that no change bits are set.
34. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
35. Test fails if Swap Completed bit (12) is not set. (9.4.4#12)
36. Test fails if PD Direction bit (22) is not opposite of what it was before SWAP.(9.4.4#12)

37. Send ClearPortPDFeature for Port Under Test with Swap Complete bit (12) set.
38. Read PD Port Status for port under test.

39. Verify that the swap was completed. [tester/compliance device]
40. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
41. Repeat test for all downstream facing ports.

11.18 Hub Intrusive Mode Negotiate Power Level REJECT_PD_REQUEST Test

Purpose	Verify REJECT_PD_REQUEST in intrusive mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11
Checklist References	

11.18.1 Test Procedure

1. Enumerate and configure the PD-capable peripheral behind PDUSB Hub Under Test.
2. Enable PD notifications by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 1 (Intrusive mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Have DUT initiate a request to renegotiate power [tester/compliance device]

Make sure hub reports this request:

7. Issue read to interrupt IN endpoint of hub.

8. Test fails if read fails (9.3.2#1)
9. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
10. Test fails if any other ports show status changes.
11. Read Port Status and Change bits for port under test.
12. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
13. Test fails if any other Port Status change bits are set.
14. Read Port Status for all ports not under test. Verify that no change bits are set.
15. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
16. Test fails if Negotiated Power Level bit (4) is not set. (9.4.4#12)
17. Verify that the power level specified in Request Data Object is the correct power level.
18. Send ClearPortPDFeature for Port Under Test with Negotiated Power Level bit (4) set.
19. Read PD Port Status for port under test.
20. Verify Negotiated Power Level bit has been cleared.

Reject request:

21. Initiate a REJECT_PD_REQUEST by sending SetFeature with the following values:
    a. wValue set to 47d (REJECT_PD_REQUEST).
    b. wIndex set to port number of attached device.
    c. wLength set to 0.
22. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
23. Verify that power request was rejected and the same power contract is in force. [tester/compliance device/analyzer]
24. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
25. Repeat test for all downstream facing ports.

11.19 Hub Intrusive Mode Swap REJECT_PD_REQUEST Test

Purpose	Verify REJECT_PD_REQUEST in intrusive mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	

Assertion References	9.4.5#11
Checklist References	

11.19.1 Test Procedure

1. Enumerate and configure the PD-capable peripheral behind PDUSB Hub Under Test.
2. Enable PD notifications by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 1 (Intrusive mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
    e. Save the value of the PD Direction bit (22).
7. Have DUT initiate a SWAP command[tester/compliance device]

Make sure hub reports this request:

8. Issue read to interrupt IN endpoint of hub.
9. Test fails if read fails (9.3.2#1)
10. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
11. Test fails if any other ports show status changes.
12. Read Port Status and Change bits for port under test.
13. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
14. Test fails if any other Port Status change bits are set.
15. Read Port Status for all ports not under test. Verify that no change bits are set.
16. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
17. Test fails if New Power Direction bit (5) is not set. (9.4.4#12)
18. Send ClearPortPDFeature for Port Under Test with New Power Direction bit (5) set.
19. Read PD Port Status for port under test.
20. Verify New Power Direction bit has been cleared.

Reject request:

21. Initiate a REJECT_PD_REQUEST by sending SetFeature with the following values:

    a. wValue set to 47d (REJECT_PD_REQUEST).
    b. wIndex set to port number of attached device.
    c. wLength set to 0.

22. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
23. Verify that swap request was not completed. [Tester/Compliance Device/analyzer]
24. Issue read to interrupt IN endpoint of hub.
25. Test fails if read fails (9.3.2#1)
26. Test fails if Port Under Test shows the change bit set. (9.3.2#1)
27. Test fails if any other ports show status changes.
28. Read Port Status for all ports not under test. Verify that no change bits are set.
29. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
30. Test fails if Swap Completed bit (12) is not set. (9.4.4#12)
31. Test fails if PD Direction bit (22) is not of the same as what it was before SWAP.(9.4.4#12)
32. Send ClearPortPDFeature for Port Under Test with Swap Complete bit (12) set.
33. Read PD Port Status for port under test.
34. Verify that the swap was not completed. [tester/compliance device]
35. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
36. Repeat test for all downstream facing ports.

11.20 Hub Intrusive Mode PORT_PD_RESET Test

Purpose	Verify PORT_PD_RESET in intrusive mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11, 9.4.6#12
Checklist References	

11.20.1 Test Procedure

1. Enumerate and configure the PDUSB Hub Under Test.

2. Notify hub that OS is PD Aware by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 1 (Intrusive mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Initiate a PD Reset by sending SetFeature with the following values:
    a. wValue set to 48d (PORT_PD_RESET).
    b. wIndex set to port number of attached device.
    c. wLength set to 0.
7. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
8. Delay to wait for reset to be complete. How much time is appropriate?
9. Verify that reset was completed. [tester/compliance device/analyzer]
10. Issue read to interrupt IN endpoint of hub.
11. Test fails if read fails (9.3.2#1)
12. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
13. Test fails if any other ports show status changes.
14. Read Port Status and Change bits for port under test.
15. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
16. Test fails if any other Port Status change bits are set.
17. Read Port Status for all ports not under test. Verify that no change bits are set.
18. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
19. Test fails if PD Reset Complete bit (6) is not set. (9.4.4#12)
20. Send ClearPortPDFeature for Port Under Test with PD Reset bit (6) set.
21. Read PD Port Status for port under test.
22. Test fails if PD Reset Complete is not 0. (assert)
23. Read PD Port Status for ports not under test. Verify that no change bits are set.
24. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
25. Test fails if SetFeature () request fails.
26. Repeat test for all downstream facing ports.

11.21 Hub Hybrid Mode SWAP Test

Purpose	Verify SWAP in hybrid mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11, 9.4.6#12
Checklist References	

11.21.1 Test Procedure

1. Enumerate and configure the PD-capable peripheral behind PDUSB Hub Under Test.
2. Notify hub that OS is PD Aware by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Hybrid mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 2 (Hybrid mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
    e. Save the value of the PD Direction bit (22).
7. Have attached device initiate a Swap. [tester/compliance device]
8. Issue read to interrupt IN endpoint of hub.
9. Test fails if read fails (9.3.2#1)
10. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
11. Test fails if any other ports show status changes.
12. Read Port Status and Change bits for port under test.
13. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
14. Test fails if any other Port Status change bits are set.
15. Read Port Status for all ports not under test. Verify that no change bits are set.
16. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.

d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)

17. Test fails if New Power Direction bit (5) is not set. (9.4.4#12)
18. Test fails if PD Direction bit (22) is not opposite of what it was before SWAP.(9.4.4#12)
19. Test fails if Swap Completed bit (12) is set. This is only for intrusive mode.
20. Send ClearPortPDFeature for Port Under Test with New Power Direction bit (5) set.
21. Read PD Port Status for port under test.
22. Test fails if New Power Direction bit (5) is not 0. (9.4.1#4)
23. Read PD Port Status for ports not under test. Verify that no change bits are set.
24. Repeat steps to swap the direction back to the original orientation.
25. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
26. Test fails if SetFeature () request fails.
27. Repeat test for all downstream facing ports.

11.22 Hub Hybrid Mode GOTO_MIN Test

Purpose	Verify GOTO_MIN in hybrid mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11
Checklist References	

11.22.1 Test Procedure

1. Enumerate and configure the PD-capable peripheral behind PDUSB Hub Under Test.
2. Enable PD notifications by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Hybrid mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 2 (Hybrid mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)

6. Make sure the hub DS port is a Provider <do we do a swap request if it is consumer?>
7. Have device initiate a GOTO_MIN request. [Tester/Compliance Device]
8. Verify that the GOTO_MIN request was accepted. [tester/compliance device can verify this was received]
9. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
10. Repeat test for all downstream facing ports.

11.23 Hub Hybrid Mode RETURN_POWER Test

Purpose	Verify RETURN_POWER in hybrid mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11
Checklist References	

11.23.1 Test Procedure

1. Enumerate and configure the compliance device behind PDUSB Hub Under Test.
2. Enable PD notifications by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Hybrid mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 2 (Hybrid mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Have device initiate RETURN_POWER request.
7. Verify that RETURN_POWER was successful [tester/compliance device]
8. Send SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).

   c. wLength set to 0.
9. Repeat test for all downstream facing ports.

11.24 Hub Hybrid Mode Request Failure Test

Purpose	Verify hub notification of failed request in hybrid mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11
Checklist References	

11.24.1 Test Procedure

1. Enumerate and configure the PD-capable peripheral behind PDUSB Hub Under Test.
2. Enable PD notifications by sending SetFeature request with the following values:
   a. wValue set to 41d (OS_IS_PD_AWARE).
   b. wIndex set to 1 (SPM is active on host).
   c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Hybrid mode by sending SetFeature request with the following values:
   a. wValue set to 42d (POLICY_MODE).
   b. wIndex set to 2 (Hybrid mode).
   c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Have device request a power level the hub cannot satisfy. [tester/compliance device]
7. Issue read to interrupt IN endpoint of hub.
8. Test fails if read fails (9.3.2#1)
9. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
10. Test fails if any other ports show status changes.
11. Read Port Status and Change bits for port under test.
12. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
13. Test fails if any other Port Status change bits are set.
14. Read Port Status for all ports not under test. Verify that no change bits are set.
15. Read PD Port Status by sending GetStatus() request with the following values:
   a. wValue set to 1.
   b. wIndex set to port under test.
   c. wLength set to 8d.

    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
16. Test fails if Request Rejected bit (9) is not set.
17. Send ClearPortPDFeature for Port Under Test with Request Rejected bit (9) set.
18. Read PD Port Status for port under test.
19. Verify bits have been cleared.
20. Return the hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
21. Repeat with device requesting a swap, if hub DS port does not support swapping power.
22. Repeat test for all downstream facing ports.

11.25 Hub Hybrid Mode PORT_PD_RESET Test

Purpose	Verify PORT_PD_RESET in hybrid mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11, 9.4.6#12
Checklist References	

11.25.1 Test Procedure

1. Enumerate and configure the PDUSB Hub Under Test.
2. Notify hub that OS is PD Aware by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Hybrid mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 2 (Hybrid mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Have device initiate hard reset. [Tester/Compliance Device]
7. Issue read to interrupt IN endpoint of hub.
8. Test fails if read fails (9.3.2#1)

9. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
10. Test fails if any other ports show status changes.
11. Read Port Status and Change bits for port under test.
12. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
13. Test fails if any other Port Status change bits are set.
14. Read Port Status for all ports not under test. Verify that no change bits are set.
15. Verify that reset was completed. [tester/compliance device/analyzer]
16. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
17. Delay to wait for reset to be complete. How much time is appropriate?
18. Test fails if PD Reset Complete bit (6) is not set. (9.4.4#12)
19. Send ClearPortPDFeature for Port Under Test with PD Reset bit (6) set.
20. Read PD Port Status for port under test.
21. Test fails if PD Reset Complete is not 0. (assert)
22. Read PD Port Status for ports not under test. Verify that no change bits are set.
23. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
24. Test fails if SetFeature () request fails.
25. Repeat test for all downstream facing ports.

11.26 Hub Local Mode SWAP Test

Purpose	Verify SWAP in non-intrusive mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11, 9.4.6#12
Checklist References	

11.26.1 Test Procedure

1. Enumerate and configure the PD-capable peripheral behind PDUSB Hub Under Test.

2. Notify hub that OS is PD Aware by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Local mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 0 (Local mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
    e. Save the value of the PD Direction bit (22).
7. Have attached device initiate a Swap. [tester/compliance device]
8. Issue read to interrupt IN endpoint of hub.
9. Test fails if read fails (9.3.2#1)
10. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
11. Test fails if any other ports show status changes.
12. Read Port Status and Change bits for port under test.
13. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
14. Test fails if any other Port Status change bits are set.
15. Read Port Status for all ports not under test. Verify that no change bits are set.
16. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
17. Test fails if New Power Direction bit (5) is not set. (9.4.4#12)
18. Test fails if PD Direction bit (22) is not opposite of what it was before SWAP.(9.4.4#12)
19. Test fails if Swap Completed bit (12) is set. This is only for intrusive mode.
20. Send ClearPortPDFeature for Port Under Test with New Power Direction bit (5) set.
21. Read PD Port Status for port under test.
22. Test fails if New Power Direction bit (5) is not 0. (9.4.1#4)
23. Read PD Port Status for ports not under test. Verify that no change bits are set.
24. Repeat steps to swap the direction back to the original orientation.
25. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
26. Test fails if SetFeature () request fails.
27. Repeat test for all downstream facing ports.

11.27 Hub Local Mode GOTO_MIN Test

Purpose	Verify GOTO_MIN in non-intrusive mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11
Checklist References	

11.27.1 Test Procedure

1. Enumerate and configure the PD-capable peripheral behind PDUSB Hub Under Test.
2. Enable PD notifications by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Local mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 0 (Local mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Make sure the hub DS port is a Provider <do we do a swap request if it is consumer?>
7. Initiate a GOTO_MIN by sending SetFeature with the following values:
    a. wValue set to 44d (GOTO_MIN).
    b. wIndex set to port number of attached device.
    c. wLength set to 0.
8. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
9. Make sure the hub DS port is a Provider <do we do a swap request if it is consumer?>
10. Have device initiate a GOTO_MIN request. [Tester/Compliance Device]
11. Verify that the GOTO_MIN request was accepted. [tester/compliance device can verify this was received]
12. Return hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
13. Repeat test for all downstream facing ports.

11.28 Hub Local Mode RETURN_POWER Test

Purpose	Verify RETURN_POWER in non-intrusive mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11
Checklist References	

11.28.1 Test Procedure

1. Enumerate and configure the compliance device behind PDUSB Hub Under Test.
2. Enable PD notifications by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 1 (SPM is active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Local mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 0 (Local mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Have device initiate RETURN_POWER request.
7. Verify that RETURN_POWER was successful [tester/compliance device]
8. Send SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
9. Repeat test for all downstream facing ports.

11.29 Hub Local Mode Request Failure Test

Purpose	Verify hub notification of failed request in local mode
Critical for Safety	No
Applies to	PDUSB hub
Description	

Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11, 9.4.6#12
Checklist References	

11.29.1 Test Procedure

1. Enumerate and configure the PDUSB Hub Under Test.
2. Notify hub that OS is PD Aware by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
3. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
4. Set hub to Local mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 0 (Local mode).
    c. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Have device request a power level the hub cannot satisfy. [tester/compliance device]
7. Issue read to interrupt IN endpoint of hub.
8. Test fails if read fails (9.3.2#1)
9. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
10. Test fails if any other ports show status changes.
11. Read Port Status and Change bits for port under test.
12. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
13. Test fails if any other Port Status change bits are set.
14. Read Port Status for all ports not under test. Verify that no change bits are set.
15. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.
    c. wLength set to 8d.
    d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
16. Test fails if Request Rejected bit (9) is not set.
17. Send ClearPortPDFeature for Port Under Test with Request Rejected bit (9) set.
18. Read PD Port Status for port under test.
19. Verify bits have been cleared.
20. Return the hub to non-intrusive mode by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
    b. wIndex set to 0 (SPM is not active on host).
    c. wLength set to 0.
21. Repeat with device requesting a swap, if hub DS port does not support swapping power.
22. Repeat test for all downstream facing ports.

11.30 Hub Local Mode PORT_PD_RESET Test

Purpose	Verify PORT_PD_RESET in non-Intrusive mode
Critical for Safety	No
Applies to	PDUSB hub
Description	
Test setup	USB30CV; PD capable peripheral
Preconditions	
Assertion References	9.4.5#11, 9.4.6#12
Checklist References	

11.30.1 Test Procedure

1. Enumerate and configure the PDUSB Hub Under Test.
2. Notify hub that OS is PD Aware by sending SetFeature request with the following values:
    a. wValue set to 41d (OS_IS_PD_AWARE).
3. wIndex set to 0 (SPM is not active on host).
4. wLength set to 0.
5. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
6. Set hub to Local mode by sending SetFeature request with the following values:
    a. wValue set to 42d (POLICY_MODE).
    b. wIndex set to 0 (Local mode).
    c. wLength set to 0.
7. Test fails if SetFeature () fails. (9.3.1#1, 9.3.1#2)
8. Have device initiate hard reset. [Tester/Compliance Device]
9. Issue read to interrupt IN endpoint of hub.
10. Test fails if read fails (9.3.2#1)
11. Test fails if Port Under Test does not show the change bit set. (9.3.2#1)
12. Test fails if any other ports show status changes.
13. Read Port Status and Change bits for port under test.
14. Test fails if bit 15 of Port Status Change bits is not set (C_PORT_PD_CHANGE). (9.3.2#2)
15. Test fails if any other Port Status change bits are set.
16. Read Port Status for all ports not under test. Verify that no change bits are set.
17. Verify that reset was completed. [tester/compliance device/analyzer]
18. Read PD Port Status by sending GetStatus() request with the following values:
    a. wValue set to 1.
    b. wIndex set to port under test.

   c. wLength set to 8d.
   d. Test fails if GetStatus() fails or does not return 8 bytes (9.4.4#1)
19. Test fails if PD Reset Complete bit (6) is not set. (9.4.4#11)
20. Send ClearPortPDFeature for Port Under Test with PD Reset bit (6) set.
21. Read PD Port Status for port under test.
22. Test fails if PD Reset Complete is not 0. (assert)
23. Read PD Port Status for ports not under test. Verify that no change bits are set.
24. Return hub to non-intrusive mode by sending SetFeature request with the following values:
   a. wValue set to 41d (OS_IS_PD_AWARE).
   b. wIndex set to 0 (SPM is not active on host).
   c. wLength set to 0.
25. Test fails if SetFeature () request fails.
26. Repeat test for all downstream facing ports.

12 MOI Assertion Coverage

Within the following section is a table of assertion coverage by MOI. The assertions covered by the MQP MOI shall be required to pass certification testing and are highlighted in GREEN. Assertions tested outside of the required set are informational only at this time but may be added to the required set as development continues. This is an informative list. If as assertion is not in this table but tested within the required test set, it may still be considered a required assertion to obtain compliance certification. Therefore, please refer to the actual test descriptions for full test coverage.

12.1 Informational vs. Required testing

Tests that ask the DUT to perform actions outside of real world application (e.g. Sending messages that are illegal in nature) are used to encourage robust design and are for informational purposes only. Failing these tests will result on warnings only and will not affect certification.

12.2 Assertions by MOI

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
3.3.2#1	BMC_PROT_SEQ_CHKCAB_P_PC_2	
5.3#1	BMC_PHY_MSG_6	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5.

Assertion	**Test from *[PDCommunicationsEngineMOI]***	**Test from *[PDDeterministicMOI]***
		TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
5.3#2	BMC_PHY_MSG_6	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
5.4#1		TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
5.4#2		TD.PD.PHY.E9.
5.5#1	BMC_PHY_MSG_6	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1.

Assertion	**Test from** ***[PDCommunicationsEngineMOI]***	**Test from** ***[PDDeterministicMOI]***
		TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
5.5#2	BMC_PHY_MSG_6	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
5.6#1	BMC_PHY_MSG_6	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
5.6#2	BMC_PHY_MSG	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
5.6#3	BMC_PHY_MSG	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
5.6.1.1#2	BMC_PHY_MSG_6	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
5.6.1.1#3	BMC_PHY_MSG_6	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3.

Assertion	**Test from *[PDCommunicationsEngineMOI]***	**Test from *[PDDeterministicMOI]***
		TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
5.6.1.1#4	BMC_PHY_MSG_6	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
5.6.1.2.1#1	BMC_PHY_MSG	TD.PD.PHY.E9. TD.PD.VNDI.E4.
5.6.1.2.1#2	BMC_PHY_MSG_2	TD.PD.PHY.E10. TD.PD.VNDI.E4.
5.6.1.2.1#3	CBL_PHY_MSG	TD.PD.PHY.E9. TD.PD.VNDI.E4.
5.6.1.2.1#4	CBL_PHY_MSG	TD.PD.PHY.E9. TD.PD.VNDI.E4.
5.6.1.2.1#5	CBL_PHY_MSG	TD.PD.PHY.E9. TD.PD.VNDI.E4.
5.6.1.2.2#1	CAB_PROT_DISCOV_1 CAB_PROT_DISCOV_3 CAB_PROT_DISCOV_5 CAB_PROT_DISCOV_7	TD.PD.PHY.E11. TD.PD.VNDI.E4.
5.6.1.2.2#3	CBL_PHY_MSG	TD.PD.PHY.E12. TD.PD.VNDI.E4.
5.6.1.2.2#4	BMC_PHY_MSG	TD.PD.PHY.E12. TD.PD.VNDI.E4.
5.6.1.2.2#5	BMC_PHY_MSG_2 BMC_PHY_MSG_4	TD.PD.PHY.E11.
5.6.1.2.2#6	BMC_PHY_MSG_2 BMC_PHY_MSG_4	TD.PD.PHY.E11.
5.6.1.2.3#1		TD.PD.VNDI.E4.
5.6.1.2.3#2	CAB_PROT_DISCOV	TD.PD.VNDI.E4.
5.6.1.2.3#3	CAB_PHY_MSG	TD.PD.PHY.E13. TD.PD.VNDI.E4.
5.6.1.2.3#4	BMC_PHY_MSG	TD.PD.VNDI.E4.
5.6.1.2.3#5	CAB_PHY_MSG	TD.PD.PHY.E14. TD.PD.VNDI.E4.
5.6.1.2.3#6	BMC_PHY_MSG	TD.PD.PHY.E14. TD.PD.VNDI.E4.
5.6.1.2.3#7	BMC_PHY_MSG_3 BMC_PHY_MSG_5	TD.PD.PHY.E13.
5.6.1.3#1	BMC_PHY_MSG_11	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
5.6.1.5#1	CAB_PHY_MSG BMC_PHY_MSG	TD.PD.PHY.E20.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
5.6.1.5#2	CAB_PHY_MSG BMC_PHY_MSG	TD.PD.PHY.E20.
5.6.1.5#3	CAB_PHY_MSG BMC_PHY_MSG	TD.PD.PHY.E20.
5.6.1.5#4	BMC_PHY_MSG_7 BMC_PHY_MSG_8	TD.PD.PHY.E20.
5.6.1.5#5	BMC_PHY_MSG_1	TD.PD.PHY.E20.
5.6.3#1	BMC_PHY_MSG_7 BMC_PHY_MSG_8 BMC_PHY_MSG_9 BMC_PHY_MSG_10	TD.PD.PHY.E1.
5.6.4#1	PROT_PROC_HR_TSTR_6	TD.PD.PHY.E16.
5.6.4#2	PROT_PROC_HR_TSTR	TD.PD.PHY.E16.
5.6.4#3	CAB_PHY_MSG BMC_PHY_MSG	TD.PD.PHY.E17.
5.6.4#4	CAB_PHY_MSG_14 BMC_PHY_MSG_16	TD.PD.PHY.E16.
5.6.4#5	CAB_PHY_MSG BMC_PHY_MSG	TD.PD.PHY.E16.
5.6.4#6		TD.PD.PHY.E16. TD.PD.LL.E4.
5.6.5#1		TD.PD.PHY.E18.
5.6.5#2	PROT_MSG_HDR_3 CAB_PHY_MSG_14	TD.PD.PHY.E18. TD.PD.PHY.E19.
5.7#1	BMC_PHY_RX_BUSIDL_2	TD.PD.PHY.E5.
5.7#2	BMC_PHY_RX_BUSIDL_3	TD.PD.PHY.E5.
5.7#3	BMC_PHY_RX_BUSIDL	TD.PD.PHY.E5.
5.8.1.1#1	CAB_PHY_RX_INT_REJ	TD.PD.PHY.E1. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8.
5.8.1.2.1#1	BMC_PHY_TX_BIT_4 CAB_PHY_TX_BIT_4	TD.PD.PHY.E4.
5.8.1.2.1#2	BMC_PHY_TX_BIT_3 CAB_PHY_TX_BIT_3	TD.PD.PHY.E4.
5.8.1.4#4	CAB_PHY_RX_BUSIDL_2	
5.8.2.1#1	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
5.8.2.3#1	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
5.8.2.5.2#1	BMC_PHY_TX_EYE_3	
5.8.2.6.4#2	BMC_PHY_RX_INT_REJ_2 BMC_PHY_RX_INT_REJ_3	
5.8.3.1#1	CAB_PHY_TERM	TD.PD.PHY.E22.
5.8.3.1#3	PHY_MSG_GEN	TD.PD.PHY.E21.
5.8.3.2.1#1	CAB_PHY_TX_EYE_3	TD.PD.PHY.E22.
5.8.3.2.1#2	CAB_PHY_TX_EYE BMC_PHY_TX_EYE	TD.PD.PHY.E22.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
5.8.3.2.1#3	CAB_PHY_TX_EYE BMC_PHY_TX_EYE	TD.PD.PHY.E22.
5.8.3.2.2#3	CAB_PHY_TX_EYE	TD.PD.PHY.E7. TD.PD.PHY.E8.
5.8.3.2.2#6	CAB_PHY_RX_INT_REJ BMC_PHY_RX_INT_REJ	TD.PD.PHY.E23.
5.8.3.2.2#7	CAB_PHY_RX_BUSIDL BMC_PHY_RX_BUSIDL	TD.PD.PHY.E23.
5.9.1#1		TD.PD.PHY.E3.
5.9.1#2		TD.PD.PHY.E3.
5.9.1#3		TD.PD.PHY.E3.
5.9.1#4		TD.PD.PHY.E3.
5.9.1#5		TD.PD.PHY.E3.
5.9.1#6		TD.PD.PHY.E3.
5.9.1#7		TD.PD.PHY.E3.
5.9.1#8		TD.PD.PHY.E3.
5.9.1.1#2		
5.9.1.2#1		TD.PD.PHY.E2.
5.9.2#1		
5.9.3#1		
5.9.4#1	BMC_PHY_TX_BIT_2 BMC_PHY_TX_EYE_2 CAB_PHY_TX_EYE_2 CAB_PHY_TX_BIT_2	TD.PD.PHY.E4.
5.9.5#1		
5.9.6#1		TD.PD.PHY.E22.
5.9.7#1	CAB_PHY_TERM BMC_PHY_TERM	TD.PD.PHY.E1.
6.2.1#1	PROT_HDR PROT_HDR_GCRC	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.2.1.1#1	PROT_HDR PROT_HDR_GCRC	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.2.1.1#2	PROT_MSG_HDR_1 PROT_MSG_HDR_16 PROT_MSG_HDR_GCRC_1 PROT_MSG_HDR_GCRC_12	TD.PD.SRC.E8. TD.PD.SRC.E10.
6.2.1.2#1	PROT_MSG_HDR_2 PROT_MSG_HDR_GCRC_2 PROT_MSG_CTRL_1 PROT_MSG_DATA_REQ_1 PROT_MSG_DATA_VDM_ID_ACK_1 PROT_MSG_DATA_VDM_SVID_INIT_1 PROT_MSG_DATA_VDM_SVID_ACK_1 PROT_MSG_DATA_VDM_MODE_INIT_1 PROT_MSG_DATA_VDM_MODE_ACK_1 PROT_MSG_DATA_VDM_ENTER_MODE_1 PROT_MSG_DATA_VDM_EXIT_MODE_1 PROT_MSG_DATA_VDM_ATT_1 PROT_MSG_DATA_SRC_CAP_1 PROT_MSG_DATA_BIST_1	TD.PD.SRC.E2. TD.PD.SRC.E8. TD.PD.SRC.E10.
6.2.1.3#1	PROT_MSG_HDR_3 PROT_MSG_HDR_4 PROT_MSG_HDR_5 PROT_MSG_HDR_GCRC_3 BMC_PHY_MSG_14	TD.PD.LL.E5.
6.2.1.3#2	PROT_MSG_HDR_6 BMC_PHY_MSG_13	TD.PD.SRC.E14.
6.2.1.4#1	PROT_MSG_HDR_11 PROT_MSG_HDR_GCRC_7	TD.PD.SRC.E2. TD.PD.SRC.E8. TD.PD.SRC.E10.
6.2.1.4#2	PROT_MSG_HDR_11 PROT_PROC_PING_1	TD.PD.SRC.E2.
6.2.1.4#3	PROT_MSG_HDR_11	TD.PD.SNK.E4.
6.2.1.4#4	PROT_MSG_HDR_11	
6.2.1.4#5	PROT_MSG_HDR_11	

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.2.1.4#7	PROT_MSG_HDR_12 PROT_MSG_HDR_GCRC_8	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.2.1.5#1	PROT_MSG_HDR_13 PROT_MSG_HDR_18 PROT_MSG_HDR_GCRC_9 PROT_MSG_HDR_GCRC_14	TD.PD.SRC.E2. TD.PD.SRC.E8. TD.PD.SRC.E10.
6.2.1.5#2	PROT_REV_NUM	
6.2.1.6#1	PROT_MSG_HDR_14 PROT_MSG_HDR_GCRC_10	TD.PD.SRC.E2. TD.PD.SRC.E8. TD.PD.SRC.E10.
6.2.1.6#2	PROT_MSG_HDR_15	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.2.1.6#3	PROT_MSG_HDR_GCRC_11	
6.2.1.7#1	PROT_MSG_HDR_GCRC_8	
6.2.1.7#2	PROT_MSG_HDR_GCRC_8 CAB_PROT_DISCOV_1 CAB_PROT_DISCOV_3 CAB_PROT_DISCOV_5 CAB_PROT_DISCOV_7	
6.2.1.8#1	PROT_MSG_HDR_17 PROT_MSG_DATA_REQ_2 PROT_MSG_DATA_VDM_ID_INIT_2 PROT_MSG_DATA_VDM_ID_ACK_2 PROT_MSG_DATA_VDM_SVID_INIT_2 PROT_MSG_DATA_VDM_SVID_ACK_2 PROT_MSG_DATA_VDM_MODE_INIT_2 PROT_MSG_DATA_VDM_MODE_ACK_2 PROT_MSG_DATA_VDM_ENTER_MODE_2 PROT_MSG_DATA_VDM_EXIT_MODE_2 PROT_MSG_DATA_VDM_ATT_2	TD.PD.SRC.E2. TD.PD.SRC.E8. TD.PD.SRC.E10.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
	PROT_MSG_DATA_SRC_CAP_2 PROT_MSG_DATA_BIST_2 PROT_MSG_DATA_VEND_1	
6.3#1	PROT_MSG_HDR_17	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.3#2	PROT_MSG_HDR_17	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.3.1#1	PROT_MSG_HDR_GCRC	TD.PD.LL.E1.
6.3.1#2	PROT_MSG_HDR_GCRC_3 BMC_PHY_MSG_12 BMC_PHY_MSG_13	TD.PD.LL.E1.
6.3.1#3	PROT_MSG_HDR_GCRC_13 PROT_PROC_GOODCRC_UUT_1	TD.PD.LL.E1.
6.3.2#1	PROT_PROC_GOTOMIN_TSTR	
6.3.3#1	PROT_PROC_REQ_TSTR	TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E11. TD.PD.SRC.E16. TD.PD.CP.E5.
6.3.3#2	PROT_PROC_SWAP_TSTR_SRC_1 PROT_PROC_SWAP_TSTR_SNK_1	TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.3.3#5	PROT_PROC_SR_TSTR_1	TD.PD.LL.E5.
6.3.3#6	PROT_PROC_REQ_TSTR PROT_PROC_SWAP_TSTR_SNK_1 PROT_PROC_SWAP_TSTR_SRC_1	TD.PD.SRC.E7.
6.3.4#1	BMC_POW_SRC_LOAD_P_PC_4	TD.PD.SRC.E12. TD.PD.SRC.E13.
6.3.4#2	PROT_SEQ_SWAP_REJ	TD.PD.VNDI.E7. TD.PD.VNDI.E8.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.3.4#7	PROT_SEQ_SWAP_REJ	TD.PD.SRC.E12. TD.PD.SRC.E13.
6.3.5.1#1	PROT_PROC_PING_2	
6.3.5.2#1	PROT_MSG_CTRL_PING_2	TD.PD.LL.E6.
6.3.6#1	PROT_PROC_SWAP_TSTR_SRC_3 PROT_PROC_SWAP_TSTR_SRC_4 PROT_PROC_SWAP_TSTR_SNK_5 PROT_PROC_SWAP_UUT_SRC_6 PROT_PROC_SWAP_UUT_SNK_5 PROT_PROC_SWAP_UUT_SNK_6	TD.PD.SRC.E9. TD.PD.SRC.E10.
6.3.7#1	PROT_PROC_GETSRCCAPS_TSTR	TD.PD.SRC.E15.
6.3.7#2	PROT_SEQ_GETSRCCAP_REJ PROT_PROC_GETSRCCAPS_TSTR	
6.3.8#1	PROT_PROC_GETSNKCAPS_TSTR	TD.PD.VNDI.E6.
6.3.8#2	PROT_PROC_GETSNKCAPS_TSTR	
6.3.10#1	PROT_PROC_SWAP_TSTR_SRC_1 PROT_PROC_SWAP_TSTR_SNK_1	TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.3.10#2	PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_TSTR_SNK_2	TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.3.13#1		TD.PD.LL.E5. TD.PD.SRC.E14. TD.PD.SNK.E9. TD.PD.SNK.E10.
6.3.13#2		TD.PD.LL.E4.
6.3.13#3		TD.PD.LL.E4.
6.3.13#4	CAB_PHY_MSG BMC_PHY_MSG	TD.PD.LL.E3.
6.3.13#5	CAB_PHY_MSG	TD.PD.SRC.E14. TD.PD.SNK.E9. TD.PD.SNK.E10.
6.4#1	PROT_MSG_HDR	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.4#2	PROT_PROC_GETSRCCAPS_TSTR_3 PROT_PROC_GETSNKCAPS_TSTR_3 BMC_PROT_SEQ_GETCAPS_1 BMC_PROT_SEQ_GETCAPS _3 BMC_PROT_SEQ_GETCAPS_1 BMC_PROT_SEQ_GETCAPS _3	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4#3		TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4#4	PROT_HDR	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4.1#1	PROT_MSG_DATA_SRC_CAP_10 PROT_PROC_SRCCAPS_UUT	TD.PD.SRC.E2.
6.4.1#2	PROT_MSG_DATA_SRC_CAP PROT_PROC_SRCCAPS_UUT	
6.4.1#3	PROT_MSG_DATA_SRC_CAP_14 PROT_PROC_SRCCAPS_UUT	TD.PD.SRC.E2.
6.4.1.1.1#1		TD.PD.VNDI.E5.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.4.1.1.1#2		TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4.1.1.1#3		TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4.1.1.1#4	PROT_MSG_DATA_SRC_CAP	
6.4.1.1.2#1		
6.4.1.2#1	PROT_MSG_DATA_SRC_CAP PROT_PROC_SRCCAPS_UUT	TD.PD.SRC.E2. TD.PD.VNDI.E5.
6.4.1.2#2	PROT_MSG_DATA_SRC_CAP_13 PROT_PROC_SRCCAPS_UUT	TD.PD.SRC.E2. TD.PD.VNDI.E5.
6.4.1.2#3	PROT_MSG_DATA_SRC_CAP PROT_PROC_SRCCAPS_UUT	TD.PD.SRC.E2. TD.PD.VNDI.E5.
6.4.1.2#4	PROT_PROC_GETSRCCAPS_TSTR	TD.PD.SRC.E2. TD.PD.VNDI.E5.
6.4.1.2#5	BMC_POW_SRC_LOAD_P_PC_2 BMC_POW_SRC_LOAD_CP_ACC_3	TD.PD.VNDI.E5.
6.4.1.2#6		TD.PD.VNDI.E5.
6.4.1.2#7	PROT_PROC_SRCCAPS_TSTR	TD.PD.VNDI.E5.
6.4.1.2#8		TD.PD.VNDI.E5.
6.4.1.2.2#1	PROT_SRC_SRC	
6.4.1.2.3#1	PROT_MSG_DATA_SRC_CAP_15	TD.PD.SRC.E2. TD.PD.VNDI.E5.
6.4.1.2.3#2	PROT_MSG_DATA_SRC_CAP_17	TD.PD.VNDI.E5.
6.4.1.2.3#3	BMC_POW_SNK_TRANS_PC POW_SNK_TRANS_C_CP	TD.PD.VNDI.E5.
6.4.1.2.3#4	PROT_MSG_DATA_SRC_CAP_9	TD.PD.SRC.E2.
6.4.1.2.3.1#1	PROT_MSG_DATA_SRC_CAP_4	TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4.1.2.3.1#2	PROT_MSG_DATA_SRC_CAP_4	TD.PD.VNDI.E7. TD.PD.VNDI.E8.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.4.1.2.3.2#1		TD.PD.VNDI.E5.
6.4.1.2.3.2#2	PROT_MSG_DATA_SRC_CAP_5	TD.PD.VNDI.E5.
6.4.1.2.3.3#1	PROT_EXT_PWR PROT_MSG_DATA_SRC_CAP_6	TD.PD.VNDI.E5.
6.4.1.2.3.4#1	PROT_MSG_DATA_SRC_CAP_7	TD.PD.VNDI.E5.
6.4.1.2.3.5#1	PROT_MSG_DATA_SRC_CAP_8	TD.PD.VNDI.E5.
6.4.1.2.3.5#2	PROT_MSG_DATA_SRC_CAP_8	TD.PD.VNDI.E5.
6.4.1.2.3.6#2	PROT_MSG_DATA_SRC_CAP_10 PROT_MSG_DATA_SRC_CAP_16	
6.4.1.2.3.6#3	PROT_MSG_DATA_SRC_CAP_10	
6.4.1.2.3.6#4	PROT_MSG_DATA_SRC_CAP_10	
6.4.1.2.4#1	PROT_MSG_DATA_SRC_CAP_19 PROT_MSG_DATA_SRC_CAP_20	
6.4.1.2.4#2	PROT_MSG_DATA_SRC_CAP_19 PROT_MSG_DATA_SRC_CAP_20	
6.4.1.2.5#1	PROT_MSG_DATA_SRC_CAP_22 PROT_MSG_DATA_SRC_CAP_23	
6.4.1.2.5#2	PROT_MSG_DATA_SRC_CAP_24	
6.4.1.3#1	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3#2	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3#3	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3#4	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3#5	PROT_MSG_DATA_SNK_CAP_13	TD.PD.VNDI.E6.
6.4.1.3#6	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3#7	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3.1#1	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1#2	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3.1#3	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1#4	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3.1.1#1	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3.1.1#2	PROT_MSG_DATA_SNK_CAP	
6.4.1.3.1.2#1	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3.1.3#1	PROT_EXT_PWR	TD.PD.VNDI.E6.
6.4.1.3.1.4#1	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3.1.5#1	PROT_MSG_DATA_SNK_CAP_8	TD.PD.VNDI.E6.
6.4.1.3.1.5#2	PROT_MSG_DATA_SNK_CAP_8	TD.PD.VNDI.E6.
6.4.1.3.2#1	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3.2#2	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3.2#3	BMC_POW_SRC_LOAD_P_PC_2 BMC_POW_SRC_LOAD_CP_ACC_3	TD.PD.VNDI.E6.
6.4.1.3.3#1	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3.3#2	PROT_MSG_DATA_SNK_CAP	TD.PD.VNDI.E6.
6.4.1.3.3#3	BMC_POW_SRC_LOAD_P_PC_2 BMC_POW_SRC_LOAD_CP_ACC_3	TD.PD.VNDI.E6.
6.4.2#1	PROT_MSG_DATA_REQ	TD.PD.SNK.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.4.2#2	PROT_MSG_DATA_REQ	TD.PD.SNK.E4.
6.4.2#3	PROT_MSG_DATA_REQ_5	TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E11. TD.PD.PC.E7.
6.4.2#4	PROT_MSG_DATA_REQ_1	TD.PD.SNK.E4.
6.4.2#5	PROT_MSG_DATA_REQ_3 PROT_MSG_DATA_REQ_8	TD.PD.SNK.E4.
6.4.2.1#1	PROT_MSG_DATA_REQ_5	TD.PD.SNK.E4.
6.4.2.2#1	PROT_MSG_DATA_REQ_13 PROT_MSG_DATA_REQ_14	TD.PD.SNK.E4.
6.4.2.3#1	PROT_MSG_DATA_REQ PROT_SEQ_CAP_MIS	TD.PD.SNK.E4.
6.4.2.3#2	PROT_MSG_DATA_REQ_5	TD.PD.SNK.E4.
6.4.2.3#3	PROT_MSG_DATA_REQ_9 PROT_MSG_DATA_REQ_12 PROT_MSG_DATA_REQ_15	TD.PD.SNK.E4.
6.4.2.3#4	PROT_MSG_DATA_REQ_16	TD.PD.SNK.E4.
6.4.2.3#5	PROT_MSG_DATA_REQ_13 PROT_MSG_DATA_REQ_14	TD.PD.SNK.E4.
6.4.2.4#1	PROT_MSG_DATA_REQ_6	
6.4.2.6#1	PROT_MSG_DATA_REQ	TD.PD.SNK.E4.
6.4.2.6#3	PROT_MSG_DATA_REQ_12 PROT_MSG_DATA_REQ_15	TD.PD.SNK.E4.
6.4.2.7#1	PROT_MSG_DATA_REQ_11 PROT_MSG_DATA_REQ_9	TD.PD.SNK.E4.
6.4.2.7#2	PROT_MSG_DATA_REQ_11 PROT_MSG_DATA_REQ_9	TD.PD.SNK.E4.
6.4.2.7#3	PROT_MSG_DATA_REQ_16 PROT_MSG_DATA_REQ_10	TD.PD.SNK.E4.
6.4.2.7#5	PROT_MSG_DATA_REQ	TD.PD.SNK.E4.
6.4.2.8#1	PROT_MSG_DATA_REQ_13 PROT_MSG_DATA_REQ_14	TD.PD.SNK.E4.
6.4.2.8#2	PROT_MSG_DATA_REQ_14	TD.PD.SNK.E4.
6.4.2.8#3	PROT_MSG_DATA_REQ	TD.PD.SNK.E4.
6.4.2.9#1	PROT_MSG_DATA_REQ_15 PROT_MSG_DATA_REQ_18	
6.4.2.9#2	PROT_MSG_DATA_REQ	
6.4.2.10#1	PROT_MSG_DATA_REQ_17	
6.4.2.10#2	PROT_MSG_DATA_REQ_16	
6.4.2.10#3	PROT_MSG_DATA_REQ_16	
6.4.2.10#4	PROT_MSG_DATA_REQ	
6.4.2.11#1	PROT_MSG_DATA_REQ_19 PROT_MSG_DATA_REQ_20	
6.4.2.11#2	PROT_MSG_DATA_REQ	
6.4.3#2		TD.PD.PHY.E2.
6.4.3#3		TD.PD.PHY.E3.
6.4.3#4		TD.PD.PHY.E2.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.4.3#7	BMC_PHY_TX_BIT_2 BMC_PHY_TX_EYE_1 BMC_PHY_TX_EYE_2 CAB_PHY_TX_EYE_1 CAB_PHY_TX_EYE_2 CAB_PHY_TX_BIT_1 CAB_PHY_TX_BIT_2 CAB_PHY_RX_BUSIDL_1	TD.PD.PHY.E4.
6.4.3#9		TD.PD.PHY.E22.
6.4.3#10	BMC_PHY_RX_BUSIDL_1 BMC_PHY_RX_INT_REJ_1 BMC_PHY_TERM_1 CAB_PHY_RX_BUSIDL_1 CAB_PHY_RX_INT_REJ_1 CAB_PHY_TERM_1	TD.PD.PHY.E1.
6.4.3.1#1		TD.PD.PHY.E2.
6.4.3.1#2		TD.PD.PHY.E2.
6.4.3.1#3	BMC_PHY_TX_BIT	TD.PD.PHY.E2.
6.4.3.1#4		TD.PD.PHY.E2.
6.4.3.1#5		TD.PD.PHY.E2.
6.4.3.2#1		TD.PD.PHY.E3.
6.4.3.2#2		TD.PD.PHY.E3.
6.4.3.2#3		TD.PD.PHY.E3.
6.4.3.2#4		TD.PD.PHY.E3.
6.4.3.2#5		TD.PD.PHY.E3.
6.4.3.2#6		TD.PD.PHY.E3.
6.4.3.2#7		TD.PD.PHY.E3.
6.4.3.2#8		TD.PD.PHY.E3.
6.4.3.2#9		TD.PD.PHY.E3.
6.4.3.3#1		TD.PD.PHY.E2.
6.4.3.4#1		
6.4.3.4#2		
6.4.3.5#1		
6.4.3.5#2		
6.4.3.6#1	PROT_PROC_BIST_TSTR CAB_PHY_TX_EYE	TD.PD.PHY.E4.
6.4.3.6#2	PROT_PROC_BIST_TSTR CAB_PHY_TX_EYE CAB_PHY_TX_BIT	
6.4.3.7#1		
6.4.3.7#2		
6.4.3.8#1		TD.PD.PHY.E22.
6.4.3.8#2		
6.4.3.9#1	BMC_PHY_RX_BUSIDL_2 BMC_PHY_TERM_1 BMC_PHY_MSG_1 CAB_PHY_RX_BUSIDL_2 CAB_PHY_TERM_1	TD.PD.PHY.E1.
6.4.3.9#2	PROT_PROC_BIST_TSTR_1	TD.PD.PHY.E1.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.4.4#1	PROT_MSG_DATA_VEND_1	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4.4#2	PROT_MSG_DATA_VEND_2	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4.4#5	PROT_MSG_DATA_VDM_ID_INIT PROT_MSG_DATA_VDM_ID_ACK	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4.4#7	PROT_MSG_DATA_VDM_ID_INIT_4 PROT_MSG_DATA_VDM_ID_ACK_4 PROT_MSG_DATA_VDM_SVID_INIT_4 PROT_MSG_DATA_VDM_SVID_ACK_4 PROT_MSG_DATA_VDM_MODE_INIT_4 PROT_MSG_DATA_VDM_MODE_ACK_4 PROT_MSG_DATA_VDM_ENTER_MODE_4 PROT_MSG_DATA_VDM_EXIT_MODE_4 PROT_MSG_DATA_VDM_ATT_4	TD.PD.VDMU.E1.
6.4.4#9		TD.PD.VDMU.E17.
6.4.4#10		TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.4.4#11	PROT_MSG_DATA_VEND	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4.4.2#1	CAB_PROT_DISCOV_1 CAB_PROT_DISCOV_3 CAB_PROT_DISCOV_5 CAB_PROT_DISCOV_7 BMC_PROT_SEQ_CHKCAB_P_PC_1	
6.4.4.2#5		TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14.
6.4.4.2#6	CAB_PHY_MSG_6	TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14.
6.4.4.2#8	PROT_MSG_DATA_VDM_ID_INIT_6 PROT_MSG_DATA_VDM_ID_INIT_9 PROT_MSG_DATA_VDM_ID_ACK_6 PROT_MSG_DATA_VDM_ID_ACK_9 PROT_MSG_DATA_VDM_SVID_INIT_6 PROT_MSG_DATA_VDM_SVID_INIT_9 PROT_MSG_DATA_VDM_SVID_ACK_6 PROT_MSG_DATA_VDM_SVID_ACK_9 PROT_MSG_DATA_VDM_MODE_INIT_6 PROT_MSG_DATA_VDM_MODE_INIT_9 PROT_MSG_DATA_VDM_MODE_ACK_6 PROT_MSG_DATA_VDM_MODE_ACK_9 PROT_MSG_DATA_VDM_ENTER_MODE_6 PROT_MSG_DATA_VDM_ENTER_MODE_9 PROT_MSG_DATA_VDM_EXIT_MODE_6 PROT_MSG_DATA_VDM_EXIT_MODE_9 PROT_MSG_DATA_VDM_ATT_6 PROT_MSG_DATA_VDM_ATT_9	TD.PD.VDMU.E1.
6.4.4.2#9	PROT_MSG_DATA_VDM	TD.PD.VDMU.E1.
6.4.4.2#10	PROT_MSG_DATA_VDM_ID_INIT_3 PROT_MSG_DATA_VDM_ID_ACK_3 PROT_MSG_DATA_VDM_SVID_INIT_3	TD.PD.VDMU.E1.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
	PROT_MSG_DATA_VDM_SVID_ACK_3 PROT_MSG_DATA_VDM_MODE_INIT_3 PROT_MSG_DATA_VDM_MODE_ACK_3	
6.4.4.2#11	PROT_MSG_DATA_VDM	TD.PD.VDMU.E1.
6.4.4.2#12	PROT_MSG_DATA_VDM	TD.PD.VDMU.E1.
6.4.4.2#13	PROT_MSG_DATA_VDM	TD.PD.VDMU.E1.
6.4.4.2.1#1	PROT_MSG_DATA_VDM_ID_INIT_3 PROT_MSG_DATA_VDM_ID_ACK_3 PROT_MSG_DATA_VDM_SVID_INIT_3 PROT_MSG_DATA_VDM_SVID_ACK_3 PROT_MSG_DATA_VDM_MODE_INIT_3 PROT_MSG_DATA_VDM_MODE_ACK_3 PROT_MSG_DATA_VDM_ENTER_MODE_3 PROT_MSG_DATA_VDM_EXIT_MODE_3 PROT_MSG_DATA_VDM_ATT_3	TD.PD.VDMU.E1.
6.4.4.2.2#1	PROT_MSG_DATA_VDM	TD.PD.VDMU.E1.
6.4.4.2.3#1	PROT_MSG_DATA_VDM_ID_INIT_5 PROT_MSG_DATA_VDM_ID_ACK_5 PROT_MSG_DATA_VDM_SVID_INIT_5 PROT_MSG_DATA_VDM_SVID_ACK_5 PROT_MSG_DATA_VDM_MODE_INIT_5 PROT_MSG_DATA_VDM_MODE_ACK_5 PROT_MSG_DATA_VDM_ENTER_MODE_5 PROT_MSG_DATA_VDM_EXIT_MODE_5 PROT_MSG_DATA_VDM_ATT_5	TD.PD.VDMU.E1.
6.4.4.2.3#2		TD.PD.VDMU.E19.
6.4.4.2.4#1	PROT_MSG_DATA_VDM_ENTER_MODE_7 PROT_MSG_DATA_VDM_EXIT_MODE_7	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.4.4.2.4#2	CAB_PHY_MSG	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4.4.2.4#5	PROT_MSG_DATA_VDM_ID_INIT_7 PROT_MSG_DATA_VDM_ID_ACK_7 PROT_MSG_DATA_VDM_SVID_INIT_7 PROT_MSG_DATA_VDM_SVID_ACK_7 PROT_MSG_DATA_VDM_MODE_INIT_7 PROT_MSG_DATA_VDM_MODE_ACK_7	TD.PD.VDMU.E1.
6.4.4.2.5#1	PROT_MSG_DATA_VDM_ID_INIT_8 PROT_MSG_DATA_VDM_ID_ACK_8 PROT_MSG_DATA_VDM_SVID_INIT_8 PROT_MSG_DATA_VDM_SVID_ACK_8 PROT_MSG_DATA_VDM_MODE_INIT_8 PROT_MSG_DATA_VDM_MODE_ACK_8 PROT_MSG_DATA_VDM_ENTER_MODE_8 PROT_MSG_DATA_VDM_EXIT_MODE_8 PROT_MSG_DATA_VDM_ATT_8	TD.PD.VDMU.E1.
6.4.4.2.5#2	PROT_MSG_DATA_VDM_ID_INIT_8 PROT_MSG_DATA_VDM_SVID_INIT_8 PROT_MSG_DATA_VDM_MODE_INIT_8 PROT_MSG_DATA_VDM_ENTER_MODE_8 PROT_MSG_DATA_VDM_EXIT_MODE_8 PROT_MSG_DATA_VDM_ATT_8	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.4.4.2.5#3	PROT_MSG_DATA_VDM	TD.PD.VDMU.E1.
6.4.4.2.5#4	PROT_MSG_DATA_VDM	TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14.
6.4.4.2.6#1	PROT_MSG_DATA_VDM	TD.PD.VDMU.E1.
6.4.4.3#1	CAB_PROT_DISCOV_1 CAB_PROT_DISCOV_3	

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
	CAB_PROT_DISCOV_5 CAB_PROT_DISCOV_7	
6.4.4.3.1#2	BMC_PROT_SEQ_CHKCAB_P_PC_1	
6.4.4.3.1#3	CAB_PROT_DISCOV_1	TD.PD.VDMU.E1.
6.4.4.3.1#4	BMC_PROT_DISCOV_1	TD.PD.VDMU.E1.
6.4.4.3.1#5	PROT_MSG_DATA_VDM_ID_INIT_3 PROT_MSG_DATA_VDM_ID_ACK_3	TD.PD.VDMU.E1.
6.4.4.3.1#6	PROT_MSG_DATA_VDM_ID_ACK_48	TD.PD.VDMU.E1.
6.4.4.3.1#7		TD.PD.VDMU.E1.
6.4.4.3.1#8		TD.PD.VDMU.E1.
6.4.4.3.1.1#1	PROT_MSG_DATA_VDM_ID_ACK	TD.PD.VDMU.E1.
6.4.4.3.1.1#2	PROT_MSG_DATA_VDM_ID_ACK_11	TD.PD.VNDI.E1.
6.4.4.3.1.1#3	PROT_MSG_DATA_VDM_ID_ACK_12	TD.PD.VNDI.E1.
6.4.4.3.1.1#4	PROT_MSG_DATA_VDM_ID_ACK_15	TD.PD.VDMU.E1.
6.4.4.3.1.1#5	PROT_MSG_DATA_VDM_ID_ACK_14	TD.PD.VNDI.E1.
6.4.4.3.1.4#1	PROT_MSG_DATA_VDM_ID_ACK_13	TD.PD.VNDI.E1.
6.4.4.3.1.4#2	PROT_MSG_DATA_VDM_ID_ACK	TD.PD.VNDI.E1.
6.4.4.3.1.4#3	PROT_MSG_DATA_VDM_ID_ACK	TD.PD.VNDI.E1.
6.4.4.3.1.4#4	PROT_MSG_DATA_VDM_ID_ACK	TD.PD.VNDI.E1.
6.4.4.3.1.4#5	PROT_MSG_DATA_VDM_ID_ACK	TD.PD.VNDI.E1.
6.4.4.3.1.4#6	PROT_MSG_DATA_VDM_ID_ACK	TD.PD.VNDI.E1.
6.4.4.3.1.4#7	PROT_MSG_DATA_VDM_ID_ACK	TD.PD.VNDI.E1.
6.4.4.3.1.6#1	PROT_MSG_DATA_VDM_ID_ACK_16	TD.PD.VNDI.E1.
6.4.4.3.1.7#1	PROT_MSG_DATA_VDM_ID_ACK_18	TD.PD.VNDI.E1.
6.4.4.3.1.7#2	PROT_MSG_DATA_VDM_ID_ACK_17	TD.PD.VDMU.E1.
6.4.4.3.1.8#1	PROT_MSG_DATA_VDM_ID_ACK_19 PROT_MSG_DATA_VDM_ID_ACK_20	TD.PD.VNDI.E1.
6.4.4.3.1.9#1	PROT_MSG_DATA_VDM_ID_ACK_49	TD.PD.VNDI.E1.
6.4.4.3.1.9#2	PROT_MSG_DATA_VDM_ID_ACK_22 PROT_MSG_DATA_VDM_ID_ACK_23 PROT_MSG_DATA_VDM_ID_ACK_25 PROT_MSG_DATA_VDM_ID_ACK_27 PROT_MSG_DATA_VDM_ID_ACK_28 PROT_MSG_DATA_VDM_ID_ACK_29 PROT_MSG_DATA_VDM_ID_ACK_30 PROT_MSG_DATA_VDM_ID_ACK_31 PROT_MSG_DATA_VDM_ID_ACK_32 PROT_MSG_DATA_VDM_ID_ACK_33 PROT_MSG_DATA_VDM_ID_ACK_34 PROT_MSG_DATA_VDM_ID_ACK_35 PROT_MSG_DATA_VDM_ID_ACK_36	TD.PD.VNDI.E1.
6.4.4.3.1.9#3	PROT_MSG_DATA_VDM_ID_ACK_24	TD.PD.VNDI.E1.
6.4.4.3.1.10#1	PROT_MSG_DATA_VDM_ID_ACK_50	TD.PD.VNDI.E1.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.4.4.3.1.10#2	PROT_MSG_DATA_VDM_ID_ACK_37 PROT_MSG_DATA_VDM_ID_ACK_38 PROT_MSG_DATA_VDM_ID_ACK_40 PROT_MSG_DATA_VDM_ID_ACK_41 PROT_MSG_DATA_VDM_ID_ACK_42 PROT_MSG_DATA_VDM_ID_ACK_43 PROT_MSG_DATA_VDM_ID_ACK_44 PROT_MSG_DATA_VDM_ID_ACK_45 PROT_MSG_DATA_VDM_ID_ACK_46 PROT_MSG_DATA_VDM_ID_ACK_47	TD.PD.VNDI.E1.
6.4.4.3.1.10#3	PROT_MSG_DATA_VDM_ID_ACK_39	TD.PD.VNDI.E1.
6.4.4.3.2#1	PROT_MSG_DATA_VDM_SVID_INIT_3	TD.PD.VDMU.E2.
6.4.4.3.2#2		TD.PD.VNDI.E2.
6.4.4.3.2#3	BMC_PROT_DISCOV_3	TD.PD.VNDI.E2.
6.4.4.3.2#4	PROT_MSG_DATA_VDM_SVID_ACK_11 PROT_MSG_DATA_VDM_SVID_ACK_12 PROT_MSG_DATA_VDM_SVID_ACK_13	TD.PD.VDMU.E2.
6.4.4.3.2#5	BMC_PROT_DISCOV CAB_PROT_DISCOV	TD.PD.VDMU.E2.
6.4.4.3.2#6	BMC_PROT_DISCOV CAB_PROT_DISCOV	TD.PD.VDMU.E2.
6.4.4.3.3#1	PROT_MSG_DATA_VDM_MODE_ACK_1	TD.PD.VDMU.E3.
6.4.4.3.3#2	CAB_PROT_DISCOV_5	TD.PD.VDMU.E3. TD.PD.VNDI.E3.
6.4.4.3.4#1	PROT_MSG_DATA_VDM_ENTER_MODE_7	TD.PD.VDMU.E4.
6.4.4.3.4#2	PROT_MSG_DATA_VDM_ENTER_MODE_1	TD.PD.VDMU.E4.
6.4.4.3.4#3	PROT_MSG_DATA_VDM_ENTER_MODE_11	TD.PD.VDMU.E4.
6.4.4.3.4#4	PROT_MSG_DATA_VDM_ENTER_MODE_1 PROT_MSG_DATA_VDM_ENTER_MODE_11	TD.PD.VDMU.E4.
6.4.4.3.4#8	CAB_PROT_DISCOV_7	TD.PD.VDMU.E4. TD.PD.VDMU.E13.
6.4.4.3.4#9	PROT_MSG_DATA_VDM_ENTER_MODE_7	TD.PD.VDMU.E4.
6.4.4.3.4#15	CAB_PROT_DISCOV_8	
6.4.4.3.4#16		TD.PD.VDMU.E18.
6.4.4.3.4#20		TD.PD.VDMU.E18.
6.4.4.3.5#1	PROT_MSG_DATA_VDM_EXIT_MODE_7	TD.PD.VDMU.E5.
6.4.4.3.5#3	PROT_MSG_DATA_VDM_EXIT_MODE_1 PROT_MSG_DATA_VDM_EXIT_MODE_11	TD.PD.VDMU.E5.
6.4.4.3.5#6		TD.PD.VDMU.E14.
6.4.4.3.6#1	PROT_MSG_DATA_VDM_ATT_7	
6.4.4.3.6#3	PROT_MSG_DATA_VDM_ATT_1 PROT_MSG_DATA_VDM_ATT_11	
6.4.4.4#1	CAB_PROT_DISCOV BMC_PROT_DISCOV	TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9.
6.4.4.4#2		TD.PD.VDMD.E1.
6.4.4.4#3	CAB_PROT_DISCOV_3	TD.PD.VNDI.E1.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.4.4.5#1		TD.PD.VDMU.E16. TD.PD.VDMD.E5.
6.4.4.5#2		TD.PD.VDMU.E16. TD.PD.VDMD.E5.
6.4.4.5#3		TD.PD.VDMU.E16. TD.PD.VDMD.E5.
6.4.4.5#4		TD.PD.VDMU.E16. TD.PD.VDMD.E5.
6.4.4.5#5		TD.PD.VDMU.E16. TD.PD.VDMD.E5.
6.4.4.5#6		TD.PD.VDMU.E16. TD.PD.VDMD.E5.
6.4.4.5#7		TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMD.E5. TD.PD.VDMD.E6.
6.5.1#1		TD.PD.LL.E2.
6.5.1#2		TD.PD.LL.E2.
6.5.1#3		TD.PD.LL.E2.
6.5.1#4		TD.PD.LL.E1. TD.PD.LL.E2.
6.5.1#5		TD.PD.LL.E1. TD.PD.LL.E2.
6.5.1#6	PROT_MSG_CTRL PROT_PROC_GOODCRC_UUT_1	TD.PD.DPD.E2.
6.5.1#7	PROT_MSG_CTRL PROT_PROC_GOODCRC_UUT_1	TD.PD.LL.E1.
6.5.2#1		TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SNK.E6.
6.5.2#2		TD.PD.SRC.E5. TD.PD.SNK.E6.
6.5.2#3		TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SNK.E6.
6.5.2#4		TD.PD.SRC.E4.
6.5.2#5	PROT_PROC_SWAP_TSTR_SNK_1	TD.PD.SRC.E7. TD.PD.SNK.E3.
6.5.2#6	PROT_MSG_CTRL	TD.PD.SRC.E7.
6.5.3.1#1	PROT_PROC_SWAP_TSTR_SNK_4 PROT_PROC_SWAP_TSTR_SRC_5 PROT_PROC_SWAP_UUT_SNK_2 PROT_PROC_SWAP_UUT_SNK_7 PROT_PROC_SWAP_UUT_SRC_2 PROT_PROC_SWAP_UUT_SRC_5 PROT_PROC_SWAP_UUT_SRC_7 PROT_PROC_REQ_TSTR	
6.5.3.1#2	PROT_MSG_CTRL_PING_2 PROT_PROC_SWAP_TSTR_SNK_4 PROT_PROC_SWAP_TSTR_SRC_5 PROT_PROC_SWAP_UUT_SNK_2 PROT_PROC_SWAP_UUT_SRC_5 PROT_PROC_SWAP_UUT_SRC_2 PROT_PROC_SWAP_UUT_SRC_7	

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
	PROT_PROC_REQ_TSTR PROT_PROC_PING_2	
6.5.3.1#3	PROT_MSG_CTRL_PING_2 PROT_PROC_SWAP_TSTR_SNK_4 PROT_PROC_SWAP_TSTR_SRC_5 PROT_PROC_SWAP_UUT_SNK_2 PROT_PROC_SWAP_UUT_SRC_7 PROT_PROC_SWAP_UUT_SRC_5 PROT_PROC_SWAP_UUT_SRC_2 PROT_PROC_REQ_TSTR	
6.5.3.2#5		TD.PD.LL.E6.
6.5.4.1#1		TD.PD.SRC.E3.
6.5.4.1#2		TD.PD.SRC.E3.
6.5.4.1#3		
6.5.4.1#4		
6.5.4.3#1		TD.PD.SRC.E1.
6.5.6.1#1	PROT_HS_PSTRANS_C_CP PROT_HS_PSTRANS_PC	TD.PD.SNK.E8.
6.5.6.1#2	PROT_HS_PSTRANS_C_CP PROT_HS_PSTRANS_PC	
6.5.6.2#1a	PROT_PROC_PSSOURCEOFFTIMER_1 PROT_PROC_PSSOURCEOFFTIMER_SWPD_1	TD.PD.CP.E1. TD.PD.CP.E2.
6.5.6.2#1b	PROT_PROC_PSSOURCEOFFTIMER_1 PROT_PROC_PSSOURCEOFFTIMER_SWPD_1	TD.PD.CP.E1. TD.PD.CP.E2.
6.5.6.2#2	PROT_PROC_PSSOURCEOFFTIMER_1 PROT_PROC_PSSOURCEOFFTIMER_SWPD_1	
6.5.6.2#3	PROT_PROC_PSSOURCEOFFTIMER_1 PROT_PROC_PSSOURCEOFFTIMER_SWPD_1 PROT_PROC_SWAP_TSTR_SNK_5	TD.PD.CP.E2.
6.5.6.3#1	PROT_PROC_PSSOURCEONTIMER_1 PROT_PROC_PSSOURCEONTIMER_SWPD_1 PROT_PROC_SWAP_TSTR_SRC_6	TD.PD.PC.E3. TD.PD.PC.E4.
6.5.6.3#2	PROT_PROC_PSSOURCEONTIMER_1 PROT_PROC_PSSOURCEONTIMER_SWPD_1	
6.5.7#1		
6.5.7#2		
6.5.7#3		
6.5.8.1#1		TD.PD.PHY.E4.
6.5.9.1#1	PROT_PROC_PSSOURCEOFFTIMER_SWPD_2 PROT_PROC_PSSOURCEONTIMER_2	
6.5.9.2#1		TD.PD.CP.E4.
6.5.9.2#2	PROT_PROC_SWAP_TSTR_SRC_7	TD.PD.CP.E4.
6.5.9.2#3		TD.PD.PC.E5.
6.5.10.2#1	PROT_PROC_HR_TSTR_1	TD.PD.SRC.E6.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.5.10.3#1		TD.PD.VDMU.E18.
6.5.11.1#1		TD.PD.VDMD.E1.
6.5.11.1#6		TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9.
6.5.11.1#7		TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9.
6.5.13#1	CAB_PROT_DISCOV_2 CAB_PROT_DISCOV_4 CAB_PROT_DISCOV_6 CAB_PROT_DISCOV_8	
6.6.1#1		TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.6.1#3	PROT_MSG_HDR_6	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.6.1#4	PROT_HDR	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.6.1#5	PROT_SEQ_GETSNKCAP_REJ	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.6.1.1#1	CAB_PHY_MSG BMC_PHY_MSG PROT_MSG_CTRL	TD.PD.LL.E5.
6.6.1.1#2	PROT_MSG_CTRL	TD.PD.LL.E2.
6.6.1.1#3	PROT_MSG_CTRL	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.6.1.2#1	PROT_MSG_CTRL	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16.

Assertion	**Test from *[PDCommunicationsEngineMOI]***	**Test from *[PDDeterministicMOI]***
		TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.6.1.2#2	PROT_MSG_CTRL CAB_PHY_MSG	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3. TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.6.1.2#3	PROT_MSG_DATA_SRC_CAP	TD.PD.PHY.E1. TD.PD.PHY.E2. TD.PD.PHY.E3. TD.PD.PHY.E4. TD.PD.PHY.E5. TD.PD.PHY.E6. TD.PD.PHY.E7. TD.PD.PHY.E8. TD.PD.PHY.E9. TD.PD.PHY.E10. TD.PD.PHY.E11. TD.PD.PHY.E12. TD.PD.PHY.E13. TD.PD.PHY.E14. TD.PD.PHY.E15. TD.PD.PHY.E16. TD.PD.PHY.E17. TD.PD.PHY.E18. TD.PD.PHY.E19. TD.PD.PHY.E20. TD.PD.PHY.E21. TD.PD.PHY.E22. TD.PD.PHY.E23. TD.PD.LL.E1. TD.PD.LL.E2. TD.PD.LL.E3. TD.PD.LL.E4. TD.PD.LL.E5. TD.PD.LL.E6. TD.PD.SRC.E1. TD.PD.SRC.E2. TD.PD.SRC.E3. TD.PD.SRC.E4. TD.PD.SRC.E5. TD.PD.SRC.E6. TD.PD.SRC.E7. TD.PD.SRC.E8. TD.PD.SRC.E9. TD.PD.SRC.E10. TD.PD.SRC.E11. TD.PD.SRC.E12. TD.PD.SRC.E13. TD.PD.SRC.E14. TD.PD.SRC.E15. TD.PD.SRC.E16. TD.PD.SNK.E1. TD.PD.SNK.E2. TD.PD.SNK.E3.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
		TD.PD.SNK.E4. TD.PD.SNK.E5. TD.PD.SNK.E6. TD.PD.SNK.E7. TD.PD.SNK.E8. TD.PD.SNK.E9. TD.PD.SNK.E10. TD.PD.SNK.E11. TD.PD.PC.E1. TD.PD.PC.E2. TD.PD.PC.E3. TD.PD.PC.E4. TD.PD.PC.E5. TD.PD.PC.E6. TD.PD.PC.E7. TD.PD.CP.E1. TD.PD.CP.E2. TD.PD.CP.E3. TD.PD.CP.E4. TD.PD.CP.E5. TD.PD.VDMU.E1. TD.PD.VDMU.E2. TD.PD.VDMU.E3. TD.PD.VDMU.E4. TD.PD.VDMU.E5. TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9. TD.PD.VDMU.E10. TD.PD.VDMU.E11. TD.PD.VDMU.E12. TD.PD.VDMU.E13. TD.PD.VDMU.E14. TD.PD.VDMU.E15. TD.PD.VDMU.E16. TD.PD.VDMU.E17. TD.PD.VDMU.E18. TD.PD.VDMU.E19. TD.PD.VDMD.E1. TD.PD.VDMD.E2. TD.PD.VDMD.E3. TD.PD.VDMD.E4. TD.PD.VDMD.E5. TD.PD.VDMD.E6. TD.PD.DPD.E2. TD.PD.VNDI.E1. TD.PD.VNDI.E2. TD.PD.VNDI.E3. TD.PD.VNDI.E4. TD.PD.VNDI.E5. TD.PD.VNDI.E6. TD.PD.VNDI.E7. TD.PD.VNDI.E8.
6.6.5#1		TD.PD.PHY.E2.
6.6.5#2		TD.PD.PHY.E2.
6.6.5#3		TD.PD.PHY.E2.
6.6.7#1	CAB_PROT_DISCOV BMC_PROT_DISCOV	TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9.
6.6.7#2	CAB_PROT_DISCOV BMC_PROT_DISCOV	TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.6.7#3		TD.PD.VDMU.E6. TD.PD.VDMU.E7. TD.PD.VDMU.E8. TD.PD.VDMU.E9.
6.7.1#1		TD.PD.LL.E3.
6.7.1#2		TD.PD.LL.E2. TD.PD.SRC.E14. TD.PD.SNK.E9. TD.PD.SNK.E10.
6.7.1#4		TD.PD.LL.E2. TD.PD.LL.E3.
6.7.1#5		TD.PD.LL.E3. TD.PD.SNK.E9. TD.PD.SNK.E10.
6.7.1#6		TD.PD.LL.E3. TD.PD.SRC.E14.
6.7.1#7		TD.PD.LL.E3. TD.PD.SRC.E14.
6.7.2.1#1		TD.PD.LL.E4.
6.7.2.1#2		TD.PD.LL.E4.
6.7.2.3#1		TD.PD.LL.E4.
6.7.3#1	CAB_PHY_MSG BMC_PHY_MSG	TD.PD.PHY.E18.
6.7.3#2	CAB_PHY_MSG BMC_PHY_MSG	TD.PD.PHY.E18.
6.7.3#3	CAB_PHY_MSG BMC_PHY_MSG	TD.PD.PHY.E18.
6.7.3#9	CAB_PHY_MSG_14	TD.PD.PHY.E18.
6.8.2.1.1#1		TD.PD.LL.E5.
6.8.2.1.3#1		TD.PD.LL.E3.
6.8.2.1.3#2		TD.PD.LL.E3.
6.8.2.1.3#3		TD.PD.LL.E3.
6.8.2.1.4#1		TD.PD.LL.E3.
6.8.2.1.4#2		TD.PD.LL.E3.
6.8.2.1.5#1		TD.PD.LL.E1. TD.PD.LL.E2.
6.8.2.1.5#2		TD.PD.LL.E1.
6.8.2.1.5#3		TD.PD.LL.E2.
6.8.2.1.6#1	PROT_MSG_HDR_GCRC_3	TD.PD.LL.E2.
6.8.2.1.6#2		TD.PD.LL.E1.
6.8.2.1.6#3		TD.PD.LL.E2.
6.8.2.1.8#1		TD.PD.LL.E2.
6.8.2.1.8#2		TD.PD.LL.E2.
6.8.2.1.8#3		TD.PD.LL.E3.
6.8.2.1.9#1		TD.PD.LL.E3.
6.8.2.1.9#2		TD.PD.LL.E3.
6.8.2.2.1#3		TD.PD.LL.E5.
6.8.2.2.2#3		TD.PD.LL.E5.
6.8.2.2.5#1		TD.PD.LL.E6.
6.8.2.3.1#2	BMC_PHY_MSG_14	

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
6.9#1		TD.PD.VDMU.E15. TD.PD.VDMD.E4.
6.9#2	PROT_PROC_GOTOMIN_TSTR_1	TD.PD.VDMU.E15. TD.PD.VDMD.E4.
6.9#3		TD.PD.VDMU.E15. TD.PD.VDMD.E4.
6.9#4		TD.PD.VDMU.E15. TD.PD.VDMD.E4.
6.9#5		TD.PD.VDMU.E15. TD.PD.VDMD.E4.
6.9#6		TD.PD.VDMU.E15. TD.PD.VDMD.E4.
6.9#7		TD.PD.VDMU.E15. TD.PD.VDMD.E4.
6.9#0		TD.PD.VDMU.E15. TD.PD.VDMD.E4.
6.9#13		TD.PD.VDMU.E15. TD.PD.VDMD.E4.
6.9#14		TD.PD.VDMU.E15. TD.PD.VDMD.E4.
6.9#15		TD.PD.PHY.E1.
6.9#16		TD.PD.VDMU.E15. TD.PD.VDMD.E4.
6.9#17		TD.PD.VDMU.E15. TD.PD.VDMD.E4.
7.1.1#5	POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.1#8	POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.2#4	PROT_PROC_SWAP_TSTR_SRC_8 PROT_PROC_SWAP_UUT_SNK_3	
7.1.3#1	PROT_MSG_DATA_SRC_CAP POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.3#2	PROT_MSG_DATA_SRC_CAP POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.3#3	PROT_MSG_DATA_SRC_CAP POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.4#1	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
7.1.4#3	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.4#4	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.4#5	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.4#6	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.4#7	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.5#2	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.5#3	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.7#1	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.9#4	POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.9#5	POW_SRC_LOAD_P_PC POW_SRC_LOAD_CP_ACC POW_SRC_LOAD_CP_REQ	
7.1.10#1	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.10#2	POW_SRC_TRANS_P_PC POW_SRC_TRANS_CP_ACC POW_SRC_TRANS_CP_REQ	
7.1.11#1	PROT_PROC_SWAP_TSTR_SNK_2 PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_UUT_SNK_3 PROT_PROC_SWAP_UUT_SRC_3	
7.1.11#2	PROT_PROC_SWAP_TSTR_SRC_8 PROT_PROC_SWAP_UUT_SNK_4 PROT_PROC_SWAP_TSTR_SNK_3	
7.1.11#4	PROT_PROC_SWAP_TSTR_SRC_8 PROT_PROC_SWAP_UUT_SNK_4	
7.1.11#5	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SNK	
7.1.12#1	POW_SNK_NOIS_C_CP	
7.1.13.1#2	POW_SNK_NOIS_C_CP	

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
7.1.13.1#3	POW_SNK_NOIS_C_CP	
7.2.2#5	PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_UUT_SNK_3	
7.2.3#1	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.2.3#2	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.2.3#3	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.2.3#4	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.2.3#5	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.2.4#2	PROT_PROC_SWAP_TSTR_SNK_6	
7.2.7#1	PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_UUT_SNK_3	
7.2.7#2	PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_UUT_SNK_3	
7.2.7#4	PROT_PROC_SWAP_TSTR_SRC_2 PROT_PROC_SWAP_UUT_SNK_4 PROT_PROC_SWAP_TSTR_SNK_2	
7.2.7#5	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SNK	
7.3.1#1	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.1#2	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.1#3	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.1#4	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.1#5	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.2#1	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.2#2	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.2#3	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.2#6	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.2#7	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.3#1	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.3#2	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.3#3	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
7.3.3#6	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.3#7	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.4#1	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.4#2	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.4#3	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.4#6	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.4#7	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.5#1	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.5#2	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.5#3	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.5#6	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.5#7	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.6#1	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	TD.PD.SNK.E11. TD.PD.PC.E7.
7.3.6#2	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	TD.PD.SNK.E11. TD.PD.PC.E7.
7.3.6#3	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	TD.PD.SNK.E11. TD.PD.PC.E7.
7.3.6#4	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.6#6	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	TD.PD.SNK.E11. TD.PD.PC.E7.
7.3.7#2	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.7#3	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.7#6	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.7#7	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.8#1	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	TD.PD.SNK.E11. TD.PD.PC.E7.
7.3.8#2	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	TD.PD.SNK.E11. TD.PD.PC.E7.
7.3.8#3	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	TD.PD.SNK.E11. TD.PD.PC.E7.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
7.3.8#5	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	TD.PD.SNK.E11. TD.PD.PC.E7.
7.3.8#6	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.9#1	PROT_PROC_SWAP_TSTR_SNK PROT_PROC_SWAP_UUT_SNK	
7.3.9#2	PROT_PROC_SWAP_TSTR_SNK PROT_PROC_SWAP_UUT_SNK	
7.3.9#4	PROT_PROC_SWAP_TSTR_SNK_2 PROT_PROC_SWAP_UUT_SNK_3	
7.3.9#6	PROT_PROC_SWAP_TSTR_SNK_3 PROT_PROC_SWAP_UUT_SNK_4	
7.3.10#1	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC	
7.3.10#2	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC	
7.3.10#3	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC	
7.3.10#4	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC_4	
7.3.10#6	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC_3	
7.3.10#7	PROT_PROC_SWAP_TSTR_SRC PROT_PROC_SWAP_UUT_SRC_4	
7.3.11#1	PROT_PROC_GOTOMIN_TSTR	
7.3.11#2	PROT_PROC_GOTOMIN_TSTR	
7.3.11#3	PROT_PROC_GOTOMIN_TSTR	
7.3.11#5	PROT_PROC_GOTOMIN_TSTR	
7.3.19#1	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
7.3.19#2	POW_SNK_TRANS_C_CP POW_SNK_TRANS_PC	
8.2.6.2#2	PROT_SEQ_SWAP_REJ	TD.PD.PC.E6.
8.3.3.2.1#4		TD.PD.CP.E4.
8.3.3.2.2#1		TD.PD.SRC.E3.
8.3.3.2.2#2		TD.PD.SRC.E3.
8.3.3.2.3#1		TD.PD.SRC.E3.
8.3.3.2.3#2		TD.PD.SRC.E3.
8.3.3.2.3#3		TD.PD.SRC.E4.
8.3.3.2.3#7		TD.PD.SRC.E3.
8.3.3.2.3#8		TD.PD.SRC.E5.
8.3.3.2.4#1		TD.PD.SRC.E7. TD.PD.SRC.E11. TD.PD.SRC.E16. TD.PD.CP.E5.
8.3.3.2.4#2		TD.PD.SRC.E7. TD.PD.SRC.E11. TD.PD.SRC.E16. TD.PD.CP.E5.
8.3.3.2.4#3		TD.PD.SRC.E7. TD.PD.SRC.E11. TD.PD.SRC.E16. TD.PD.CP.E5.

Assertion	**Test from** ***[PDCommunicationsEngineMOI]***	**Test from** ***[PDDeterministicMOI]***
8.3.3.2.5#1		TD.PD.SRC.E7. TD.PD.SRC.E11. TD.PD.SRC.E16. TD.PD.CP.E5.
8.3.3.2.5#2		TD.PD.SRC.E7. TD.PD.SRC.E11. TD.PD.SRC.E16. TD.PD.CP.E5.
8.3.3.2.5#3		TD.PD.SRC.E9. TD.PD.SRC.E11. TD.PD.SRC.E16. TD.PD.CP.E5.
8.3.3.2.5#4		TD.PD.SRC.E9. TD.PD.SRC.E11. TD.PD.SRC.E16. TD.PD.CP.E5.
8.3.3.2.6#7		TD.PD.SRC.E15.
8.3.3.2.12#1		TD.PD.SRC.E15.
8.3.3.2.12#2		TD.PD.SRC.E15.
8.3.3.3#1		TD.PD.LL.E6.
8.3.3.3.1#2		TD.PD.SNK.E4.
8.3.3.3.3#1		TD.PD.SNK.E2.
8.3.3.3.3#2		TD.PD.SNK.E3.
8.3.3.3.4#1		TD.PD.SNK.E3.
8.3.3.3.4#2		TD.PD.SNK.E3.
8.3.3.3.4#3		TD.PD.SNK.E3. TD.PD.SNK.E6.
8.3.3.3.5#1		TD.PD.SNK.E3.
8.3.3.3.5#2		TD.PD.SNK.E3.
8.3.3.3.5#3		TD.PD.SNK.E5. TD.PD.SNK.E6.
8.3.3.3.5#4		TD.PD.SNK.E7. TD.PD.SNK.E8.
8.3.3.3.5#7		TD.PD.SNK.E6.
8.3.3.3.6#1		TD.PD.SNK.E7. TD.PD.SNK.E8.
8.3.3.3.7#9		TD.PD.LL.E6.
8.3.3.4.1.1#1		TD.PD.LL.E3.
8.3.3.4.1.1#2		TD.PD.LL.E3.
8.3.3.4.1.1#3		TD.PD.LL.E3.
8.3.3.4.1.1#4		TD.PD.LL.E3.
8.3.3.4.1.1#5		TD.PD.LL.E4.
8.3.3.4.1.1#6		TD.PD.LL.E4.
8.3.3.4.1.2#1		TD.PD.LL.E5.
8.3.3.4.1.2#2		TD.PD.LL.E5.
8.3.3.4.1.2#3		TD.PD.LL.E5.
8.3.3.4.1.2#4		TD.PD.LL.E5.
8.3.3.4.2.1#1		TD.PD.LL.E3.
8.3.3.4.2.1#2		TD.PD.LL.E3.
8.3.3.4.2.1#3		TD.PD.LL.E3.
8.3.3.4.2.1#4		TD.PD.LL.E3.
8.3.3.4.2.1#5		TD.PD.LL.E4.
8.3.3.4.2.1#6		TD.PD.LL.E4.
8.3.3.4.2.1#7		TD.PD.LL.E4.
8.3.3.4.2.2#1		TD.PD.LL.E5.
8.3.3.4.2.2#2		TD.PD.LL.E5.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
8.3.3.4.2.2#3		TD.PD.LL.E5.
8.3.3.4.2.2#4		TD.PD.LL.E5.
8.3.3.6.3.1.3#1		TD.PD.PC.E1.
8.3.3.6.3.1.3#2		TD.PD.PC.E1.
8.3.3.6.3.1.4#1		TD.PD.PC.E1.
8.3.3.6.3.1.5#1		TD.PD.PC.E2. TD.PD.CP.E3.
8.3.3.6.3.1.5#2		TD.PD.PC.E2. TD.PD.CP.E3.
8.3.3.6.3.1.6#1		TD.PD.PC.E2.
8.3.3.6.3.1.6#2		TD.PD.PC.E3. TD.PD.PC.E4.
8.3.3.6.3.1.6#5		TD.PD.PC.E4.
8.3.3.6.3.1.7#7		TD.PD.PC.E1.
8.3.3.6.3.2.3#1		TD.PD.CP.E2.
8.3.3.6.3.2.3#2		TD.PD.CP.E2.
8.3.3.6.3.2.4#1		TD.PD.CP.E2.
8.3.3.6.3.2.4#2		TD.PD.CP.E2.
8.3.3.6.3.2.6#1		TD.PD.CP.E3.
8.3.3.6.3.2.6#3		TD.PD.CP.E3.
8.3.3.6.3.2.6#4		TD.PD.CP.E3.
8.3.3.6.3.2.7#1		TD.PD.CP.E2.
8.3.3.6.3.2.7#6		TD.PD.CP.E2.
8.3.3.8.1.1#1		TD.PD.VDMU.E1.
8.3.3.8.1.1#2		TD.PD.VDMU.E1.
8.3.3.8.1.1#3		TD.PD.VDMU.E1.
8.3.3.8.1.2#1	BMC_PHY_MSG_6 BMC_PHY_MSG_12	TD.PD.VDMU.E1.
8.3.3.8.1.2#2	BMC_PHY_MSG_6 BMC_PHY_MSG_12	TD.PD.VDMU.E1.
8.3.3.8.1.3#1		TD.PD.VDMU.E1.
8.3.3.8.1.3#2		TD.PD.VDMU.E1.
8.3.3.8.2.1#1		TD.PD.VDMU.E2.
8.3.3.8.2.1#2		TD.PD.VDMU.E2.
8.3.3.8.2.1#3		TD.PD.VDMU.E2.
8.3.3.8.2.2#1		TD.PD.VDMU.E2.
8.3.3.8.2.2#2		TD.PD.VDMU.E2.
8.3.3.8.2.3#1		TD.PD.VDMU.E2.
8.3.3.8.2.3#2		TD.PD.VDMU.E2.
8.3.3.8.3.1#1		TD.PD.VDMU.E3.
8.3.3.8.3.1#2		TD.PD.VDMU.E3.
8.3.3.8.3.1#3		TD.PD.VDMU.E3.
8.3.3.8.3.2#1		TD.PD.VDMU.E3.
8.3.3.8.3.2#2		TD.PD.VDMU.E3.
8.3.3.8.3.3#1		TD.PD.VDMU.E3.
8.3.3.8.3.3#2		TD.PD.VDMU.E3.
8.3.3.8.4.1#1		TD.PD.VDMU.E4.
8.3.3.8.4.1#2		TD.PD.VDMU.E4.

Assertion	Test from *[PDCommunicationsEngineMOI]*	Test from *[PDDeterministicMOI]*
8.3.3.8.4.1#3		TD.PD.VDMU.E4.
8.3.3.8.4.2#1		TD.PD.VDMU.E4.
8.3.3.8.4.2#2		TD.PD.VDMU.E4.
8.3.3.8.4.3#1		TD.PD.VDMU.E4.
8.3.3.8.4.3#2		TD.PD.VDMU.E4.
8.3.3.8.5.1#1		TD.PD.VDMU.E5.
8.3.3.8.5.1#2		TD.PD.VDMU.E5.
8.3.3.8.5.1#3		TD.PD.VDMU.E5.
8.3.3.8.5.2#1		TD.PD.VDMU.E5.
8.3.3.8.5.2#2		TD.PD.VDMU.E5.
8.3.3.8.5.3#1		TD.PD.VDMU.E5.
8.3.3.8.5.3#2		TD.PD.VDMU.E5.
8.3.3.9.1.1#1		TD.PD.VDMD.E1.
8.3.3.9.1.1#2	BMC_PHY_MSG_6 BMC_PHY_MSG_12	TD.PD.VDMD.E1.
8.3.3.9.1.1#3		TD.PD.VDMD.E1.
8.3.3.9.1.2#1		TD.PD.VDMD.E1.
8.3.3.9.1.2#2		TD.PD.VDMD.E1.
8.3.3.9.3.1#1		TD.PD.VDMD.E1.
8.3.3.9.3.1#3		TD.PD.VDMD.E2. TD.PD.VDMD.E3.
8.3.3.10.2.2#1	BMC_PHY_MSG_6 BMC_PHY_MSG_12	TD.PD.VDMU.E2.
8.3.3.10.2.2#2	BMC_PHY_MSG_6 BMC_PHY_MSG_12	TD.PD.VDMU.E2.
8.3.3.10.2.3#1	BMC_PHY_MSG_6 BMC_PHY_MSG_12	TD.PD.VDMU.E2.
8.3.3.10.2.3#2	BMC_PHY_MSG_6 BMC_PHY_MSG_12	TD.PD.VDMU.E2.
8.3.3.10.7.1#1		TD.PD.LL.E5.
8.3.3.10.7.1#2		TD.PD.LL.E5.
8.3.3.10.7.1#3		TD.PD.LL.E5.
8.3.3.10.11.1#5	BMC_PHY_MSG_6 BMC_PHY_MSG_12	TD.PD.VDMU.E2.
8.3.3.11.3.4#1		TD.PD.PHY.E4.