

# USB Power Delivery Compliance Test Specification

---

***Revision:***                ***Q3, 2026 OR***  
***Release date:***        ***May 31<sup>st</sup>, 2026***

Copyright © 2016-2026, 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.

## Editors

Dmitriy Berchanskiy

Intel Corporation

## Contributors

Chee Lim Nge

Yoon Lee

Abel Astley

Mario Pasquali

Hao Sun

Chuck Trefts

Tim Wei

Nikhil Acharaya

Balamurugan Manialagan

Prasannakumar Marikunte Nagaraju

Mahesh Nakala

Krishnamurthy Parameshwara

Jian Zen Khaw

Rajaraman Venkatachalam

Harshavardhan Reddy Gonugunta

Naresh Botsa

Nathalie Ballot

Dmitriy Berchanskiy

Tyler Joe

Chetan Kopalle

Jason Shen

Mike Micheletti

Deric Waters

Biju Erayamkot Panayamthatta

Shafiuddin Mohammed

Bob Dunstan

Sten Carlsen

Diane Lenox

Apple

Apple

Ellisys

Ellisys

Ellisys

Ellisys

Ellisys

Granite River Labs

Granite River Labs

Granite River Labs

Granite River Labs

Granite River Labs

Granite River Labs

Granite River Labs

Granite River Labs

Granite River Labs

STMicroelectronics

Intel Corporation

Teledyne LeCroy

Teledyne Lecroy

Teledyne Lecroy

Teledyne LeCroy

Texas Instruments

Texas Instruments

Texas Instruments

USB Implementers Forum.

USB Implementers Forum.

USB Implementers Forum

## Table of Contents

Editors .....	3
Contributors .....	3
Revision History .....	10
Revision .....	11
Release Date.....	11
Description.....	11
Q4 2025.....	11
October 12, 2025.....	11
PPS Tests_Update.....	11
EPR capable field Changes.....	11
TEST.PD.PS.SNK.1 .....	11
TEST.PD.PS.SRC.2_SPR_AVS.....	11
TEST.PD.VDM.CBL.1 .....	11
TEST.PD.PROT.SRC3.5 .....	11
TEST.PD.PS.SRC.SNK_w_InvariantPDO_N_CHanges .....	11
TEST.PD.VDM.SRC.1 .....	11
TEST.PD.PS.SRC.2_PPS_AVS .....	11
TEST.PD.EPR.SNK3.11.....	11
Q1 2026.....	12
1/27/2026 .....	12
Error_Recovery_ECR.....	12
Wait_Message_Response_ECR.....	12
TEST.PD.PROT.SNK3.3.....	12
TEST.PD.PROT.SNK3.2.....	12
TEST.PD.PROT.SRC3.5 .....	12
TEST.PD.PROT.SRC3.4 .....	12
TEST.PD.PROT.SRC3.14_ECR .....	12
COMMON.CHECK.PD.2 .....	12
COMMON.PROC.BU.5 .....	12
COMMON.PROC.BU.2 .....	12
COMMON.PROC.BU.6 .....	12
COMMON.PROC.BU.9 .....	12
COMMON.PROC.BU.12 .....	12

PROT.SNK3.6_and_7.....	12
PROT.SNK.9 .....	12
PROT.SNK.8 .....	12
PROT.SNK.3 .....	12
PROT.SNK.14 .....	12
COMMON.PROC.BU.8 .....	12
COMMON.CHECK.PD.11 .....	12
SenderResponseTimeoutCheck .....	12
Q2 2026.....	12
3/9/2026 .....	12
TEST.PD.PROT.SRC3.14_ECR .....	12
COMMON.PROC.PD.18_ECR.....	12
TEST.PD.VDM.SNK.5.....	12
Q3 2026.....	12
TBD.....	<b>Error! Bookmark not defined.</b>
TEST.PD.PS.SNK.2 .....	12
COMMON.CHECK.PD3.4 .....	12
SenderResponseTimeoutCheck .....	12
nCTSWaitCount_output .....	12
COMMON_CHECK_PD3.2_EnterMode .....	12
TEST.PD.VDM_SRC1_SNK1 .....	12
1 Introduction.....	13
1.1 Overview .....	13
1.2 Terms and Abbreviations .....	14
1.2.1 The types of UUTs .....	14
1.3 Keywords .....	15
1.3.1 Check .....	15
1.3.2 PD2 mode.....	15
1.3.3 PD3 mode.....	15
1.3.4 Revision Alignment .....	15
1.4 Test Requirements.....	15
1.4.1 VCONN Levels for Testing Cable Markers or VPDs .....	15
1.4.2 Test Condition Codes.....	15
1.4.3 “Supports Structured VDMs” .....	18

1.4.4	Error Recovery .....	18
1.4.5	V <sub>BUS</sub> Measurement – Point of Reference.....	18
1.4.6	Tester Default Behavior .....	19
1.4.7	Reporting Missing Message Bytes .....	19
1.4.8	Reporting When Test Cannot be Completed.....	19
1.4.9	Test Cable Requirement and Dealing with VCONN.....	19
1.4.10	Vendor ID and Product IDs used in testing.....	19
2	Preparation Procedure .....	20
	This section has been deprecated. ....	20
3	Common Checks, Bring-up Procedures, and Common Procedures .....	20
3.1	Common Checks .....	20
3.1.1	PD2 and PD3 Modes .....	20
3.1.2	PD3 Mode.....	38
3.2	Bring-up Procedures.....	43
3.2.1	PD2 and PD3 Modes.....	43
3.3	Common Procedures .....	57
3.3.1	PD2 and PD3 Modes.....	57
3.3.2	PD3 Mode .....	68
4	Physical Layer Specific Tests.....	76
4.1	Test Procedures for All UUTs.....	80
4.1.1	PD3 Mode .....	80
4.2	Test Procedures for Source and Sink Capable UUTs .....	96
4.2.1	PD3 Mode .....	96
5	Protocol Specific Tests .....	99
5.1	Test Procedures for All UUTs.....	99
5.1.1	PD2 and PD3 Modes.....	99
5.1.2	PD3 Mode .....	113
5.2	Test Procedures for Source and Sink Capable UUTs .....	129
5.2.1	PD2 and PD3 Modes.....	129
5.2.2	PD3 Mode .....	129
5.3	Test Procedures for Source Capable UUT .....	139
5.3.1	PD2 and PD3 Modes.....	139
5.3.2	PD3 Mode .....	151
	TEST.PD.PROT.SRC3.15 <i>Alert Response Extended Alert.</i> .....	164

5.4	Test Procedures for Sink Capable UUT .....	165
5.4.1	PD2 and PD3 Modes.....	165
5.4.2	PD3 Mode .....	178
5.5	Structured VDM Test Procedures for Source Capable UUT .....	185
5.5.1	PD2 and PD3 Modes.....	185
5.6	Structured VDM Test Procedures for Sink Capable UUT .....	187
5.6.1	PD2 and PD3 Modes.....	187
5.7	Structured VDM Test Procedures for Cable Plug UUT.....	194
5.7.1	PD2 and PD3 Modes.....	194
5.7.2	PD3 Mode .....	200
6	Power Supply Specific Tests .....	202
6.1	Test Procedures for Source Capable UUT .....	204
6.1.1	PD2 and PD3 Modes.....	204
6.2	Test Procedures for Sink Capable UUT .....	222
6.2.1	PD2 and PD3 Modes.....	222
7	Extended Power Range Tests.....	233
7.1	Test Procedure for EPR Source Capable UUT.....	233
7.1.1	PD3 Mode .....	233
7.1.2	PD3 Mode – EPR_Power_Supply Specific Tests.....	258
7.2	Test Procedure for EPR Sink Capable UUT .....	263
7.2.1	PD3 Mode.....	263
8	Fast Role Swap Tests .....	280
8.1	Fast Role Swap Tests – Initial Source.....	280
8.1.1	PD3 Mode.....	280
8.2	Fast Role Swap Tests – Initial Sink .....	289
8.2.1	PD3 Mode.....	289
9	USB4.....	298
9.1	USB4 Tests – Data Reset .....	298
9.1.1	PD3 Mode.....	298
9.2	USB4 Tests – Enter USB4 .....	307
9.2.1	PD3 Mode.....	307
9.3	USB4 Tests – Cable Response .....	317
9.3.1	PD3 Mode.....	317
Appendix A: Requirements for Compliance Test Results .....		319

Appendix B: XML Output Example.....	320
Appendix C: Power Disconnection for FRS Initial Source Testing .....	323
AC Mains / Wall Power Switch Box .....	324
DC Switch Box.....	325
Cable interfaces and BOM .....	326
Appendix D: $V_{BUS}$ measurements .....	327
D.1 Filter Function .....	327
D.2 Coefficients .....	328
Appendix F: Timing table.....	329
tCtsSnkRequest Calculations .....	330
tCtsSrcAccept Calculations.....	331
Appendix E. Slew Rate Measurements.....	332
Appendix F. Signal In Presence of Noise .....	332
Appendix G Managed Capability Port Test Flow.....	333

## TABLES

Table 1 UUT Types .....	14
Table 2 Cable Types .....	14
Table 3 - VCONN Test Voltages.....	15
Table 4 Test Condition Codes for Ports.....	16
Table 5 Test Conditions Codes for Cables or VPDs.....	18
Table 6 Max_Current_Field Truth Table.....	29
Table 7 Src_PDO_Max_PowerN Truth Table.....	29
Table 8 PDP .....	30
Table 9 Receiver Test Signaling Table.....	76
Table 10 Rev 2 Port - Number of VDOs Expected .....	190
Table 11 Rev 2 Cable - Number of VDOs Expected.....	191
Table 12 Rev 3 Port Number of VDOs Expected.....	191
Table 13 Cable - Number of VDOs Expected .....	191
Table 14 PDO Sequence .....	202
Table 15 EPR PDO Sequence.....	202
Table 16 Sink Capabilities Extended Message .....	209
Table 17 Sink_Capabilities_Extended Message with Sink Maximum PDP = 100W .....	220
Table 18 Components .....	327
Table 19 Timing Table & Calculations .....	329
Figure 1: using mid-level to find the reference unit interval .....	76
Figure 2: Example of producing eye diagram.....	77
Figure 3: Noise injection generator and calibration fixture setup.....	77
Figure 4: Two-tone noise calibration.....	78
Figure 5: AWG measurement waveforms .....	79



Figure 6 Voltage and Current Tolerances .....	209
Figure 7 FRS Testing Concept.....	281
Figure 8 $V_{CONN}$ measurements during Data Reset.....	304
Figure 9 FRS Connection Diagram.....	324
Figure 10 Schematic of the Switch Box for disconnecting AC Mains / Wall Power .....	324
Figure 11 schematic of the standard switch box for disconnecting DC power .....	325
Figure 12 Plug and Socket for disconnecting DC power.....	326
Figure 13 Wiring diagram.....	326
Figure 14 $t_{CtsSnkRequest}$ Calculations.....	330
Figure 15 $t_{CtsSrcAccept}$ Calculation.....	331
Figure 16 Signal Timing in Presence of Noise .....	332

## Revision History

Revision	Release Date	Description
Q4 2025	October 12, 2025	<a href="#">PPS Tests Update</a> <a href="#">EPR capable field Changes</a> <a href="#">TEST.PD.PS.SNK.1</a> <a href="#">TEST.PD.PS.SRC.2 SPR AVS</a> <a href="#">TEST.PD.VDM.CBL.1</a> <a href="#">TEST.PD.PROT.SRC3.5</a> <a href="#">TEST.PD.PS.SRC.SNK_w_InvariantPDO_N_Changes</a> <a href="#">TEST.PD.VDM.SRC.1</a> <a href="#">TEST.PD.PS.SRC.2 PPS AVS</a> <a href="#">TEST.PD.EPR.SNK3.11</a> <a href="#">TEST_SnkCapExt</a> <a href="#">TEST.PD.PS.SNK.1</a>

Q1 2026	1/27/2026	<a href="#">Error Recovery ECR</a> <a href="#">Wait Message Response ECR</a> <a href="#">TEST.PD.PROT.SNK3.3</a> <a href="#">TEST.PD.PROT.SNK3.2</a> <a href="#">TEST.PD.PROT.SRC3.5</a> <a href="#">TEST.PD.PROT.SRC3.4</a> <a href="#">TEST.PD.PROT.SRC3.14 ECR</a> <a href="#">COMMON.CHECK.PD.2</a> <a href="#">COMMON.PROC.BU.5</a> <a href="#">COMMON.PROC.BU.2</a> <a href="#">COMMON.PROC.BU.6</a> <a href="#">COMMON.PROC.BU.9</a> <a href="#">COMMON.PROC.BU.12</a> <a href="#">PROT.SNK3.6 and 7</a> <a href="#">PROT.SNK.9</a> <a href="#">PROT.SNK.8</a> <a href="#">PROT.SNK.3</a> <a href="#">PROT.SNK.14</a> <a href="#">COMMON.PROC.BU.8</a> <a href="#">COMMON.CHECK.PD.11</a> <a href="#">SenderResponseTimeoutCheck</a>
Q2 2026	3/9/2026	<a href="#">TEST.PD.PROT.SRC3.14 ECR</a> <a href="#">COMMON.PROC.PD.18 ECR</a> <a href="#">TEST.PD.VDM.SNK.5</a>
Q3 2026	05/31/2026	<a href="#">TEST.PD.PS.SNK.2</a> <a href="#">COMMON.CHECK.PD3.4</a> <a href="#">SenderResponseTimeoutCheck</a> <a href="#">nCTSWaitCount output</a> <a href="#">COMMON CHECK PD3.2 EnterMode</a> <a href="#">TEST.PD.VDM SRC1 SNK1</a> <a href="#">TEST.PD.PROT.SRC3.5</a>

## 1 Introduction

This document specifies USB-IF compliance tests for a USB PD3.1 device. This test specification covers USB-IF testing for compliance with the requirements in Chapters 5-8 and 10 in the USB PD2.0 and PD3.1 Specifications.

The following documents as well as any other tests mandated by the USB-IF may be applicable to a USB PD3.0 device:

- USB Type-C Functional Test Specification
- USB Type-C and PD Source Power Test Specification
- USB Command Verifier Compliance Test Specification
- USB 2.0 Electrical Test Specification

This document is organized as follows:

- Chapter 2 describes the common test procedures and checks that would be repeated in the tests described in the subsequent chapters.
- Chapter 3 describes tests that primarily cover Chapter 5 requirements in the USB PD2.0 and PD3.1 Specifications.
- Chapter 4 describes tests that primarily cover Chapters 6, 8 and 10 requirements in the USB PD2.0 and PD3.1 Specifications.
- Chapter 5 describes tests that primarily cover Chapter 7 requirements in the USB PD2.0 and PD3.1 Specifications.

### 1.1 Overview

The Tester runs tests applicable to the UUT in Chapter 3-5. The UUT applicability is determined by the VIF fields, and it is provided at the beginning of each subsection e.g. “The tests in this subsection are applicable to UUT with VIF field VIF\_Product\_Type set to Port”.

The Tester performs checks on one Port of the UUT at a time, and the Port shall be capable of achieving Maximum Power that port can offer per VIF.

The Tester runs the common checks and procedures as described in Chapter 2 for all tests.

The applicability of common check for a given PD message is provided in Check Applicability e.g. Message Header check (COMMON.CHECK.PD.2) is applicable to “all PD messages except *GoodCRC*”.

The applicability of common procedure is provided in Procedure Applicability e.g. unless otherwise specified, the Tester runs this procedure (COMMON.PROC.PD.2) whenever receiving *Get\_Source\_Cap* Message from the UUT. The exceptions are provided either in the test steps or Test Specific Tester Behavior.

The Tester runs all the steps as described in the test (in Chapter 3-5) and the common procedure (in Section 2.3) unless the condition for “the test stops here” is satisfied, the test does not stop when a check fails.

If the test passes, the Tester continues to monitor communication, Voltage, Current and Power levels on CC, V<sub>CONN</sub> and V<sub>BUS</sub> for *tCtsAfterLastPoint* after the last point of a test has concluded and shall mark the test as “Fail” in case if the behavior of The UUT is not in accordance with the PD specification. If the “*tCtsAfterLastPoint*” failure is not directly related to the last point of the test, the Tester would provide description of the failure in the test comment.

During evaluation of Voltage, Current and Power measurements the Tester shall take into account UUT's accuracy and precision prescribed by the PD specification for a current mode of operation (PPS, SPR, etc.)

In summary:

- Bring-up Procedures in Section 2.2 are used to start the tests in Chapter 3-5
- Common Checks in Section 2.1 are used in all the tests depending on the Check Applicability
- Common Procedures in Section 2.3 are used in all the tests depending on the Procedure Applicability and exceptions provided in the test.
- The test does not stop until the last step or the condition for “the test stops here” is satisfied.

## 1.2 Terms and Abbreviations

Tester is the instrument used for running the test procedures and checks.

Unit Under Test (UUT) is the product undergoing the test procedures and checks.

### 1.2.1 The types of UUTs

The types of UUTs to be tested using this Specification are defined by parameters in the VIF.

The following table indicates the relationship with those parameters. The highlighted cells represent the minimum set of cells for UUT type identification.

Table 1 UUT Types

	VIF Product Type	PD Port Type	Product Type	Captive Cable	Captive Cable Is eMarked	Certification Type	SOP P Capable
DRP	Port Product	DRP	-	Any	Any	Any	Any
Provider Only	Port Product	Provider Only	-	Any	Any	Any	Any
Consumer Only	Port Product	Consumer Only	-	Any	Any	Any	Any
Provider/Consumer	Port Product	Provider/Consumer	-	Any	Any	Any	Any
Consumer/Provider	Port Product	Consumer/Provider	-	Any	Any	Any	Any
Free Cable	Cable	-	Passive Cable, Active Cable	-	-	EndProduct	YES
eMarked Captive Cable*	Port Product	Any except eMarker	Passive Cable, Active Cable	YES	YES	Any	YES
VPD	Port Product	eMarker	Vconn Powered Device	YES	-	Any	YES
Si-Only eMarker	Cable	-	Passive Cable, Active Cable	-	-	Silicon	YES

\* This UUT type will also match one of other five upper rows.

Table 2 Cable Types

	VIF Product Type	Captive Cable is eMarked	Product Type
Passive Cable	Cable (1)	Yes	3

Active Cable	Cable (1)	Yes	4
--------------	-----------	-----	---

\*(VIF Product Type || Captive Cable is eMarked) && Product Type

### 1.3 Keywords

#### 1.3.1 Check

Check is a keyword used to indicate requiring the Tester to determine whether the UUT has failed to comply the mandatory requirement in the base specification.

#### 1.3.2 PD2 mode

The test is running in PD2 mode if the Tester as a Sink sends the first *Request* Message with Message Header Bits 7...6 (Specification Revision) set to 01b.

The test is running in PD2 mode if the Tester as a Source sends the *Source\_Capabilities* and *Discover Identity* Commands with Message Header Bits 7...6 (Specification Revision) set to 01b.

#### 1.3.3 PD3 mode

The test is running in PD3 mode if the Tester as a Sink sends the first *Request* Message with Message Header Bits 7...6 (Specification Revision) set to 10b.

The test is running in PD3 mode if the Tester as a Source sends the *Source\_Capabilities* and *Discover Identity* Commands with Message Header Bits 7...6 (Specification Revision) set to 10b.

#### 1.3.4 Revision Alignment

‘Revision alignment’ means after the revision interoperability check has been done, as described in 6.2.1.1.5 of USB PD specification.

### 1.4 Test Requirements

#### 1.4.1 VCONN Levels for Testing Cable Markers or VPDs

Cable Markers and VPDs shall be tested at the highest and lowest VCONN voltages that the Type-C Specification allows. In certain tests they are also tested at the mid-level voltage. These voltages are specified in Table 1. When referring to these voltages, this Test Specification uses the names given in Table 1, rather than repeating the actual voltage values.

The values of VCONN used for VPDs are derived directly from the Type-C Specification. The values of VCONN used for Cable Markers is derived from the Type-C Specification with an additional +/- 0.25V ground offset caused by current being supplied from the source.

Table 3 - VCONN Test Voltages

	Cable Markers	VPDs
Maximum	5.75V	5.5V
Mid-level	4.25V	4.25V
Minimum	2.75V	3.0V

#### 1.4.2 Test Condition Codes

The majority of the tests in this Test Specification are repeated more than once under different Test Conditions. A list of the applicable Test Conditions appears at the start of each test. This section defines the codes for these conditions to be used in that list and also in the XML results.

e.g: `<condition conditionID=" Rev3ChkdSrc">`

#### 1.4.2.1 Common Test Condition

Checks that are performed as part of Common Checks, Bring-up Procedures or Common Procedures may occur during any Test Condition, but are only reported once, and are therefore not given a specific Test Condition Code. During the XML report the ‘condition’ element shall be defined: `<condition conditionID="">`.

#### 1.4.2.2 Test Conditions for Ports

Test Conditions for Ports are a combination of some or all of the following:

- The PD Revision for which the Port is being tested.
- Whether the Tester is set to support Unchunked Extended Messages
- Whether the Port is being tested as an initial Source or as an initial Sink
- Whether the test cable has been flipped at the UUT end

For example, the Code ‘Rev3ChkdSrc’ means that the Port is being tested for PD Revision 3, with the Tester set not to support Unchunked Extended Messages, and as a Source.

Table 4 shows the complete list of Test Conditions used for Ports.

*Table 4 Test Condition Codes for Ports*



Test Condition Code	PD Revision	Tester Supports Unchunked Extended messages	Source/Sink	FR_Swap Condition	Comment
Rev2Src	2	-	Source	-	
Rev2Snk	2	-	Sink	-	
Rev3ChkdSrc	3	No	Source	-	
Rev3ChkdSnk	3	No	Sink	-	
Rev3UnchkdSrc	3	Yes	Source	-	Test is also applicable if VIF parameter Unchunked_Extended_Messages_Supported is set to false
Rev3UnchkdSnk	3	Yes	Sink	-	Test is also applicable if VIF parameter Unchunked_Extended_Messages_Supported is set to false
CabFlippedSrc*	3	-	Source	-	
CabFlippedSnk*	3	-	Sink	-	
Rev3ChkdSrc5VAC	3	No	Source	Start with a 5V contract, then interrupt the AC Mains/Wall Power using Appendix B, Fig B-1	
Rev3ChkdSrc5VDC	3	No	Source	Start with a 5V contract, then interrupt the DC supply using Appendix B, Fig B-2	
Rev3ChkdSrcHiVAC	3	No	Source	Start with Highest Fixed PDO contract, then interrupt the AC Mains/Wall Power using Appendix B, Fig B-1	
Rev3ChkdSrcHiVDC	3	No	Source	Start with Highest Fixed PDO contract, then interrupt the DC supply using Appendix B, Fig B-2	

\*This code is only used when checking the Eye Diagram with the cable flipped at the UUT end

#### 1.4.2.3 Test Conditions for Cable Markers or VPDs

Test Conditions for Free Cables, Captive Cables\*, Si-Only eMarker or VPDs are a combination of some or all of the following:

- The PD Revision for which the UUT is being tested
- The VCONN Level from Table 1 which is in use

- Which end of the cable is being tested (not for VPDs or Si-Only eMarker)

For example, the Code ‘Rev2VcMaxEnd2’ means that the Cable Marker is being tested for PD Revision 2, with VCONN set to Maximum, and End 1 of the Cable is being tested.

*\*Please note that the simplified term ‘Captive Cable’ is used in Test Conditions tables to mean ‘Captive Cables Presenting Ra’.*

**Table 5** Test Conditions Codes for Cables or VPDs shows the complete list of Test Conditions used for Cable Markers or VPDs.

Table 5 Test Conditions Codes for Cables or VPDs

Test Condition Code	PD Revision	Vconn Level	Free Cable		VPD, Captive Cable, Si-Only eMarker
			End1	End2	
Rev2VcMinEnd1	2	Minimum	✓		✓
Rev2VcMaxEnd1	2	Maximum	✓		✓
Rev2VcMinEnd2	2	Minimum		✓	
Rev2VcMaxEnd2	2	Maximum		✓	
Rev3VcMinEnd1	3	Minimum	✓		✓
Rev3VcMidEnd1	3	Mid-level	✓		✓
Rev3VcMaxEnd1	3	Maximum	✓		✓
Rev3VcMinEnd2	3	Minimum		✓	
Rev3VcMidEnd2	3	Mid-level		✓	
Rev3VcMaxEnd2	3	Maximum		✓	

#### 1.4.3 “Supports Structured VDMs”

There are three VIF parameters that determine whether a Port UUT, using SOP signaling, supports Structured VDMs.

- Responds\_To\_Discov\_SOP\_DFP
- Responds\_To\_Discov\_SOP\_UFP
- Attempts\_Discov\_SOP

Throughout this document the phrase “Supports Structured VDMs Is YES” will be used to indicate that one of the appropriate VIF parameters is set to YES, and “Supports Structured VDMs is NO” if all of the appropriate VIF parameters are set to NO.

#### 1.4.4 Error Recovery

There are a number of places in this document where a check is required as to whether Error Recovery takes place within a valid timeframe. For the sake of clarity, Error Recovery is deemed to have taken place when the CC line voltage falls outside the ‘connect’ range, as a result of either Rp or Rd being released by the UUT. Subsequently the port under test stops driving V<sub>BUS</sub> and/or V<sub>CONN</sub>. Error Recovery is associated with *ErrorRecovery* state described in the latest version of the USB Type-C Cable and Connector Specification.

#### 1.4.5 V<sub>BUS</sub> Measurement – Point of Reference

For a captive cable, V<sub>BUS</sub> shall be measured at the Tester connector, otherwise it shall be measured at the UUT connector.

#### 1.4.6 Tester Default Behavior

Unless explicitly defined within a test procedure, the Tester should operate as though it were an actual PD and Type-C port. This includes following all state diagrams and transitions in a manner consistent with the latest specification. For example, *GoodCRC* Messages are to be sent automatically with a proper *MessageID*, even though that is not typically stated, and Soft and *Hard Resets* should be issued when appropriate.

#### 1.4.7 Reporting Missing Message Bytes

As the PD Specification develops, it may happen that one or more bytes get added to a message, and on UUTs that do not yet implement the later version of the specification, there will be a missing byte. This has the potential to create multiple failure points, resulting from both the incorrect message length and missing fields. To avoid this redundancy, checks should not be performed on fields that are entirely absent. Instead, reporting of those failures should be limited to the Common Checks (for example COMMON.CHECK.PD.10) where the correct number of bytes for the message in question is checked.

#### 1.4.8 Reporting When Test Cannot be Completed.

When a Test cannot be completed because, for example, the UUT unexpectedly removes  $V_{BUS}$  when behaving as a Source, or performs an unexpected Error Recovery, the Checks that could not be made shall return the result 'FAIL' together with the failure explanation being 'Test could not be completed'. This is to avoid an ambiguous result such as 'NOT TESTED'.

#### 1.4.9 Test Cable Requirement and Dealing with VCONN

When VIF field *Captive\_Cable* = NO, the UUT shall be connected with a special Test Cable. The Test Cable defined for the purposes of running tests is not a standard USB Cable. It differs in the following respects:

- It has no Cable eMarker, as this is emulated within the tester (see NOTE below).
- It has two CC lines.

Additional requirements are that:

- It is 48V capable.
- It is 5A capable.

This implies that VCONN is never applied to the Test Cable, as this would conflict with any VCONN applied by the UUT. However, it is necessary to apply VCONN in the case of a UUT with Captive Cable. In either case the timing associated with the virtual VCONN is simulated by the Tester.

Any reference to the Tester driving VCONN in this specification should be in this context.

**NOTE:** Unless otherwise defined in a specific test or section of tests, the Tester shall emulate an eMarker within the cable by presenting  $R_a$  on the second CC line and sending Discover Identity responses as defined in COMMON.PROC.PD.7

#### 1.4.10 Vendor ID and Product IDs used in testing.

The VID is 0x1A0A, this VID is used in multiple specs for testing. The PIDs presented by the PID/VID corresponds with the following test modes.

Table x Product IDs used in testing.

PID	Test Mode	Used in
0x0101	TEST_SEO_NAK	Electrical testing
0x0102	TEST_J	Electrical testing
0x0103	TEST_K	Electrical testing
0x0104	TEST_PACKET	Electrical testing
0x0105	RESERVED	Electrical testing
0x0106	HS_HOST_PORT_SUSPEND_RESUME	Electrical testing
0x0107	SINGLE_STEP_GET_DEV_DESC	Electrical testing
0x0108	SINGLE_STEP_SET_FEATURE	Electrical testing
0x0200	TTST_CONFIG	OTG
0x0201	Unknown Device Not Supporting HNP	OTG
0x0202	Unknown Device Supporting HNP	OTG
0x0301	<i>Cable</i>	PD
0x0302	Source/Sink	PD

## 2 Preparation Procedure

This section has been deprecated.

## 3 Common Checks, Bring-up Procedures, and Common Procedures

This chapter describes the Common Checks, Bring-up Procedures, and Common Procedures that would be repeated in the tests described in the subsequent chapters.

### 3.1 Common Checks

#### 3.1.1 PD2 and PD3 Modes

The common procedures in this subsection are applicable when the test is running in PD2 and PD3 modes.

##### *COMMON.CHECK.PD.1 Check Preamble*

**Description:** The Tester performs the physical layer checks on the Preamble sent by the UUT.

**Check Applicability:** All PD messages.

The Tester performs the following checks on the Preamble of all PD messages

**[COMMON.CHECK.PD.1#1]**

- The check fails if the first bit of the Preamble does not start at least *tInterFrameGap* after the end of the last bit of the preceding message.
- The check fails if the Preamble is not a 64-bit sequence of alternating '0's and '1's. (In practice the last 60 bits are checked as there is some uncertainty about detecting the initial bits.)
- The check fails if the Preamble does not end with a single “one”.

*COMMON.CHECK.PD.2 Check Message Header*

**Description:** The Tester performs the protocol checks on the Message Header sent by the UUT.

**Check Applicability:** All PD messages except *GoodCRC*.

The Tester performs the following checks on the Message Header of all PD messages except *GoodCRC* and Returned BIST Counters: **[COMMON.CHECK.PD.2#1]**

1. Bit 7...6: (Specification Revision)
  - a. For messages sent by the UUT as initial Source after detaching, *Hard Reset*, or *Error Recovery*, prior to Revision Alignment this check fails if this field is not set to 10b. This applies to:
    - i. Messages sent over SOP prior to the first *Request* from the Tester and
    - ii. Messages sent over SOP' prior to the first *Request* over SOP or the first *Discover Identity* response over SOP' from the Tester.
  - b. For all other messages, the check fails if the Specification Revision differs from the revision specified by the present Test Condition.
2. Bits 14...12: Number of Data Objects
  - a. When the Extended field is set to zero, the check fails if Number of Data Objects field does not match the number of 32-bit Data Objects that follow the Message Header
  - b. When the Extended bit (in the Message Header) is set to 1b and Chunked bit (in the Extended Message Header) is set to 1b, the check fails if the size in bytes of the payload following the Message Header is not 4 times the Number of Data Objects
  - c. When the Extended bit (in the Message Header) is set to 1b and Chunked bit (in the Extended Message Header) is set to 0b, the check fails if Number of Data Objects field is not set to 000b.
  - d. Data messages:
    - i. A *Request* Message: Number of Data Objects = 1
    - ii. A *Battery\_Status* Message: Number of Data Objects = 1
    - iii. An *Alert* Message: Number of Data Objects = 1
    - iv. A *Get\_Country\_Info* Message: Number of Data Objects = 1
    - v. An *Enter\_USB* Message: Number of Data Objects = 1
    - vi. An *EPR\_Request* Message: Number of Data Objects = 2
3. Bits 11...9: *MessageID*. The check fails if the value is not 000b under the following conditions:
  - a. The first message on each SOP\* type after a *Hard Reset* signaling was sent or received .
  - b. The message is a *Soft\_Reset* Message

- c. The first message on the same SOP\* type after receiving *Soft\_Reset*.
- d. The first message on SOP after a successful Power Role Swap
- e. The first message on SOP after a successful Fast Role Swap
- f. The first message on each SOP\* type upon initial entry to Attached state.
- g. The first message on each SOP' and SOP'' after a *Cable Reset* or *Data Reset*
- 4. Bits 11...9: *MessageID*. If *MessageID* is incremented for a valid retry scenario as described in the test procedures, fail the check. (On retry all message fields must remain the same from the previously received message.) Else, fail the check if *MessageID* is not incremented.
- 5. Bits 11...9: *MessageID*. Except for conditions as mentioned above, the check fails if *MessageID* is incremented by three comparing to the previously received message.
- 6. Bit 8: Port Power Role (SOP packet type only)
  - a. The check fails if Port Power Role does not represent the UUT port power role.
  - b. .
  - c. For a *Request* Message, the check fails if Port Power Role is not Sink.
  - d. The check fails if the Port Power Role is not Sink for the first *PS\_RDY* during a Power Role Swap
  - e. The check fails if the Port Power Role is not Source for the second *PS\_RDY* during a Power Role Swap
- 7. Bit 8: Cable Plug (SOP' and SOP''), the Tester checks that it follows the rule:
  - a. 0b Message originated from a DFP or UFP
  - b. 1b Message originated from a Cable Plug or VP
- 8. Bit 5: Port Data Role (SOP packet type only). The check fails if Port Data Role does not represent the UUT port data role.
- 9. Bit 5: Port Data Role (SOP' and SOP'' packet types). The check fails if Port Data Role is not set to zero.

#### *COMMON.CHECK.PD.3 Check GoodCRC*

Description: The Tester performs the protocol checks on the *GoodCRC* Message received by the Tester.

Check Applicability: All *GoodCRC* Messages except in TEST.PD.PHY.ALL.5 Receiver Interference Rejection

The Tester performs the following checks on the reception of *GoodCRC* Message:

- 1. The check fails if the *GoodCRC* is not received or received after *tTransmit* max has expired. The delay is measured between the last bit of the EOP of Tester sent message and the first bit of the Preamble of the UUT sent *GoodCRC* Message.  
[COMMON.CHECK.PD.3#1]
- 2. Perform the following checks on the Message Header of *GoodCRC* Message:  
[COMMON.CHECK.PD.3#2]
  - a. Bit 15 (Reserved for PD2, Extended for PD3). The check fails if this field is not set to 0b
  - b. Bits 11...9: *MessageID*. The check fails if *MessageID* is not the same as the value in the preceding message sent by the Tester.
  - c. Bit 8: Port Power Role (SOP packet type only). The check fails if Port Power Role does not represent the UUT port power role.
  - d. Bit 8: Cable Plug (SOP' and SOP''), the Tester checks that it follows the rule:

- i. If VIF\_Product\_Type is 0 (Port Product) and Captive\_Cable\_is\_eMarked as NO, the check fails if the field is not set to 0.
  - ii. If VIF\_Product\_Type is 1 (Cable), the check fails if the field is not set to 1.
- e. Bits 7...6 (Specification Revision). The check fails if this field is not set to 01b when the message this *GoodCRC* is responding to has a Specification Revision field set to 01b.
- f. Bits 7...6 (Specification Revision). The check fails if this field is set to 11b.
- g. Bit 5: Port Data Role (SOP packet type only). The check fails if Port Data Role does not represent the UUT port data role.
- h. Bit 5: Port Data Role (SOP' and SOP'' packet types). The check fails if Port Data Role is not set to zero.
- i. Bits 4...0 (Message Type) is set to 0001b.

#### *COMMON.CHECK.PD.4 Check Atomic Message Sequence Response*

**Description:** The Tester performs additional protocol checks on response messages as part of the Atomic Message Sequence (AMS).

**Check Applicability:** All PD response messages when an AMS has started

The Tester performs the following check when an AMS has started:

1. As part of an AMS, the check fails if the UUT does not respond to a message (sent by the Tester) within *tReceiverResponse* max or *tVDMReceiverResponse* max from the last bit of the EOP of the *GoodCRC* Message sent [COMMON.CHECK.PD.4#1]. The following are the exceptions:
  - a. *PS\_RDY* Message as part of Power Negotiation AMS
  - b. *PS\_RDY* Message as part of Power Role Swap AMS
  - c. *PS\_RDY* Message as part of VCONN Swap AMS
  - d. *Security\_Response* Message as part of Security AMS
  - e. *Enter\_Mode* Command request as part Enter Mode AMS
  - f. *Exit\_Mode* Command request as part Exit Mode AMS
  - g. *Not\_Supported* Message as part of chunked request

#### *COMMON.CHECK.PD.5 Check Unexpected Messages and Signals*

**Description:** The Tester performs additional protocol checks on unexpected messages sent by the UUT.

**Check Applicability:** All PD messages.

The check fails if the UUT unexpectedly removes V<sub>BUS</sub> when behaving as a Source or performs an unexpected Error Recovery while the UUT behaves as a Sink or a Source.

[COMMON.CHECK.PD.5#5].

Unless specifically mentioned, the check fails if the *Soft Reset* Message (using SOP packet) is sent by the UUT during a test [COMMON.CHECK.PD.5#1].

Unless specifically mentioned, the check fails if the UUT sends a *Hard Reset* or a *Cable Reset* signal [COMMON.CHECK.PD.5#2].

The following messages sent by the UUT are unexpected during a test when the Tester is in *PE\_SRC\_Ready*, *PE\_SNK\_Ready* or *PE\_CBL\_Ready* state (Policy Engine States as described in the base specification), and the check fails. This does not apply to such messages when they are



expected in these states as a normal part of the protocol (examples are *Not\_Supported* received after an *Alert*, *Firmware\_Update\_Request* or *Security\_Request*). [COMMON.CHECK.PD.5#4]:  
In PD2 Mode:

- *GoodCRC*
- *BIST*
- *Sink\_Capabilities*
- *Accept*
- *Reject*
- *Wait*
- *PS\_RDY*
- Structured VDM response (ACK, NAK, BUSY)

In PD3 Mode:

- *GoodCRC*
- *BIST*
- *Sink\_Capabilities*
- *Accept*
- *Reject*
- *Wait*
- *PS\_RDY*
- Structured VDM response (ACK, NAK, BUSY)
- 
- *Not\_Supported*
- *Battery\_Status*
- *Source\_Capabilities\_Extended*
- *Status*
- *Battery\_Capabilities*
- *Manufacturer\_Info*
- *PPS\_Status*
- *Country\_Info*
- *Country\_Codes*
- *Sink\_Capabilities\_Extended*
- *Data\_Reset\_Complete*
- *Source\_Info*
- *Revision*
- *Request*

#### COMMON.CHECK.PD.6 Check Control Message

**Description:** The Tester performs additional protocol checks on all Control messages sent by the UUT.

**Check Applicability:** Control messages including *GoodCRC*, , *Accept*, *Reject*, *PS\_RDY*, *Get\_Source\_Cap*, *Get\_Sink\_Cap*, Protocol Error, *DR\_Swap*, *PR\_Swap*, VCONN\_Swap, *Wait*, *Soft\_Reset*, *Get\_Source\_Info*, *Get\_Revision Message*.

The Tester performs the following additional check on all Control messages

[COMMON.CHECK.PD.6#1]:

1. The check fails if the Number of Data Object field in the Header is not zero.



2. The check fails if the number of bytes of the payload following the header is not zero.

#### *COMMON.CHECK.PD.7 Check Source\_Capabilities Message*

Description: The Tester performs additional protocol checks on all *Source\_Capabilities* Messages sent by the UUT.

Depending on UUT's internal power policies, the UUT may offer power that is lower than in the VIF PD\_Power\_As\_Source field and as a result the PDOs will not match PDOs specified in the VIF. For those types of UUTs the Tester would run a reduced set of checks. On the other hand, if the UUT offers full power all the time regardless of internal or external conditions, the Tester would run the full set of checks in this test. To differentiate between two types of UUTs, the Tester uses VIF Parameter Has\_Invariant\_PDOs:

- If VIF Parameter Has\_Invariant\_PDOs is set to Y – run all checks.
- If VIF Parameter Has\_Invariant\_PDOs is set to N
  - o and if current Source Capabilities match those in the VIF, run all checks (see note below)
  - o else, if current Source Capabilities do not match those in the VIF, run all checks except those marked FULL CAPABILITY ONLY

NOTE: There are some tests where the Tester will attempt to induce UUTs with Has\_Invariant\_PDOs = N to send their full capabilities. In those instances, the FULL CAPABILITY ONLY checks will be performed in accordance with the conditions above.

Check Applicability: the *Source\_Capabilities* Message

Perform the following checks on all *Source\_Capabilities* Messages:

1. Without receiving a SOP' *Discover Identity* Command from the UUT, the check fails if VIF field Captive\_Cable = No and the UUT sends *Source\_Capabilities* offering for source currents greater than 3A and/or voltages greater than 20V.  
[COMMON.CHECK.PD.7#1]
2. For sink, the check fails if the UUT initiates source capability message without receiving the *Get\_Source\_Cap* Message from the Tester. [COMMON.CHECK.PD.7#2]
3. **FULL CAPABILITY ONLY:** The Tester checks the Number of Data Objects field in the Message Header [COMMON.CHECK.PD.7#3]
  - a. In PD3 mode, the check fails if the Number of Data Objects field in the Message Header is not equal to VIF field Num\_Src\_PDOs.
  - b. In PD2 mode, the check fails if the Number of Data Objects field in the Message Header is not equal to VIF field Num\_Src\_PDOs minus number of PPS PDOs and/or SPR AVS PDOs depending on what the UUT supports.
4. For the first PDO, the Tester checks the consistency [COMMON.CHECK.PD.7#4]
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 1b if VIF field PD\_Port\_Type is set to Consumer/Provider, Provider/Consumer or DRP.
  - c. B28 (USB Suspend Supported) set to 1b if VIF field USB\_Suspend\_May\_Be\_Cleared = NO
  - d. B27 (Unconstrained Power) matches VIF field Unconstrained\_Power
  - e. B26 (USB Communications Capable) matches VIF field USB\_Comms\_Capable
  - f. B25 (Dual-Role Data) set to 1b if VIF field DR\_Swap\_To\_UFP\_Supported or DR\_Swap\_To\_DFP\_Supported = YES

- g. Bit 23 (EPR Mode Capable) matches VIF field EPR\_Supported\_as\_Src in PD3 mode.
- h. For the *Source\_Capabilities* Message before the first response from the Tester
  - i. Bit 24 (Unchunked Extended Messages Supported) matches VIF field Unchunked\_Extended\_Messages\_Supported
  - ii. Bits 22 (Reserved) set to 0b.
- i. For the *Source\_Capabilities* Message after the Tester has sent the *Request* Message
  - i. Bit 24 (Unchunked Extended Messages Supported) matches VIF field Unchunked\_Extended\_Messages\_Supported if the test is running in PD3 mode.
  - ii. Bit 22 (Reserved) set to 0b if the test is running in PD3 mode.
  - iii. Bits 24...22 (Reserved) set to 000b if the test is running in PD2 mode.
- j. B21...20 (Peak Current) matches VIF field Src\_PDO\_Peak\_Current1
- k. B19...10 (Voltage) set to 5V.
- l. B9...0 (Maximum Current)

**FULL CAPABILITY ONLY:** For UUT as a Source:

- 1. If Src\_PDO\_Max\_current1 ≤ 3A, Src\_PDO\_Max\_Current1
- 2. If Src\_PDO\_Max\_current1 > 3A:
  - a. The current is limited to 3A if:
    - i. an emulated cable is not e-marked, or it is 3A capable only and
    - ii. the Captive\_Cable = NO
  - b. Otherwise, the current shall be Src\_PDO\_Max\_Current1

**NOTE:** For the truth table of *Max\_Current\_Field*, which is Src\_PDO\_Max\_current1, see **Table 6** Max\_Current\_Field Truth Table

- 5. If it is not the first Fixed Supply PDO, check that B29...22 is set to zero [COMMON.CHECK.PD.7#5]
- 6. If it is a Programmable Power Supply APDO, check that B26...25, B16 and B7 are set to zero [COMMON.CHECK.PD.7#6]
- 7. If it is an Adjustable Voltage Supply APDO, check that B25...20 are set to zero [COMMON.CHECK.PD.7#13]
- 8. The Tester compares all PDOs to **PDP** See **Table 8** and check that they meet the requirements of the Power Rules. [COMMON.CHECK.PD.7#7]
  - a. Fixed Variable PDO
    - i. Check the maximum current is RoundUp (**PDP**/Voltage) or RoundDown (**PDP**/Voltage) to the nearest 10mA.
    - ii. Check that all the normative voltages are supported by UUT based on the PDP as per the SPR Normative Voltages and Minimum Currents Table in the PD3 specification.
    - iii. Check that the maximum voltage does not exceed 20V.
  - b. Battery PDO
    - i. Check that the maximum power is less than or equal to PDP.
  - c. SPR AVS APDO

- i. Max\_Current\_Field:
        - 1. B9...0 (15V-20V) range: Check the maximum current matches 20V Fixed PDO B9...0 (Maximum Current).
        - 2. B19...10 (9V-15V) range: Check that the maximum current matches 15V Fixed PDO B9...0 (Maximum Current).
    - d. SPR PPS APDO
      - i. Check that the maximum current is RoundDown (PDP/Prog Voltage) to the nearest 50mA.
      - ii. Check that all the normative SPR PPS voltages are supported by UUT based on the **PDP** as per SPR Programmable Power Supply PDOs and APDOs PDP Table in the PD3 specification if SPR PPS ADO exists.
      - iii. Check that the maximum voltage does not exceed 21V.
      - iv. Check that the maximum voltage and minimum voltage are as per the nominal voltages mentioned in SPR Programmable Power Supply Voltage Ranges Table in the PD3 specification.
  - 9. **FULL CAPABILITY ONLY:** For each PDO, the Tester checks consistency of: [COMMON.CHECK.PD.7#8]
    - a. Src\_PDO\_Supply\_TypeN
    - b. If Fixed Supply PDO:
      - i. Src\_PDO\_VoltageN
      - ii. Src\_PDO\_Peak\_CurrentN
      - iii. Max\_current\_field:
        - For UUT as a Source:
          - a. If Src\_PDO\_Max\_currentN <= 3A,  
Src\_PDO\_Max\_CurrentN
          - b. If Src\_PDO\_Max\_currentN > 3A:
            - i. The current is limited to 3A if:
              - 1. emulated cable is not e-marked, or it is 3A capable only and
              - 2. the Captive\_Cable = NO
            - ii. Otherwise, the current shall be Src\_PDO\_Max\_CurrentN
- NOTE: For the truth table of **Max\_Current\_Field**, which is **Src\_PDO\_Max\_currentN**, see **Table 6** Max\_Current\_Field Truth Table
- c. If Variable Supply PDO:
    - i. Src\_PDO\_MinVoltageN
    - ii. Src\_PDO\_MaxVoltageN
    - iii. Max\_current\_Field:
      - For UUT as a Source:
        - a. If Src\_PDO\_Max\_currentN <= 3A,  
Src\_PDO\_Max\_CurrentN
        - b. If Src\_PDO\_Max\_currentN > 3A:
          - i. The Current is limited to 3A if:
            - 1. emulated cable is not e-marked, or it is 3A capable only and
            - 2. the Captive\_Cable = NO

- ii. Otherwise, the current shall be  
Src\_PDO\_Max\_CurrentN

**NOTE:** For the truth table of Max\_Current\_Field, which is

**Src\_PDO\_Max\_currentN**, see [Table 6](#) Max\_Current\_Field Truth Table

d. If Battery Supply PDO:

- i. Src\_PDO\_MinVoltageN
- ii. Src\_PDO\_MaxVoltageN
- iii. Src\_PDO\_Max\_PowerN field:

For UUT as a Source:

- a. If  $(\text{Src\_PDO\_Max\_PowerN} / \text{Src\_PDO\_MinVoltageN}) \leq 3\text{A}$ , Src\_PDO\_Max\_PowerN
- b. If  $(\text{Src\_PDO\_Max\_PowerN} / \text{Src\_PDO\_MinVoltageN}) > 3\text{A}$ :

- i. The current is limited to  $(\text{Src\_PDO\_Max\_PowerN} / \text{Src\_PDO\_MinVoltageN})$  if
  - 1. emulated cable is not e-marked, or it is 3A capable only and
  - 2. the Captive\_Cable = NO
- ii. Otherwise, the current shall be

$\text{Src\_PDO\_Max\_PowerN} / \text{Src\_PDO\_MinVoltageN}$

**NOTE:** For the truth table of Src\_PDO\_Max\_PowerN field, see [Table 7](#) Src\_PDO\_Max\_PowerN Truth Table

e. If Adjustable Voltage Supply PDO:

- i. B27...26 (Peak Current) matches VIF field Src\_PDO\_Peak\_Current1
- ii. Max\_current\_field:
  - 1. B9...0 (15V-20V) range: Follow the steps 9.b.iii (Fixed Supply PDO) and the Tester checks the Src\_PDO\_Max\_CurrentN (20V Fixed PDO B9...0, Maximum Current).
  - 2. B19...10 (9V – 15V) range: Follow the steps 9.b.iii (Fixed Supply PDO) and the Tester checks the Src\_PDO\_Max\_CurrentN (15V Fixed PDO B9...0, Maximum Current)

**NOTE:** For the truth table of Max\_Current\_Field, which is

**Src\_PDO\_Max\_CurrentN**, see [Table 6](#) Max\_Current\_Field Truth Table

f. If Programmable Power Supply PDO:

- i. Src\_PDO\_MinVoltageN
- ii. Src\_PDO\_MaxVoltageN
- iii. Max\_current\_field:

For UUT as a Source:

- a. If  $\text{Src\_PDO\_Max\_currentN} \leq 3\text{A}$ ,  
Src\_PDO\_Max\_CurrentN
- b. If  $\text{Src\_PDO\_Max\_currentN} > 3\text{A}$ :
  - i. The current is limited to 3A if
    - 1. emulated cable is not e-marked, or it is 3A capable only and
    - 2. the Captive\_Cable = NO

- ii. Otherwise, the current shall be  
Src\_PDO\_Max\_CurrentN

**NOTE:** For the truth table of **Max\_Current\_Field**, which is  
**Src\_PDO\_Max\_currentN**, see **Table 6** Max\_Current\_Field Truth Table

10. The Tester checks PDOs following the first one, are in the correct order: Fixed PDOs in increasing Voltage sequence, Battery PDOs in increasing Minimum Voltage sequence, Variable PDOs in increasing Minimum Voltage sequence, and finally Programmable Power Supply APDOs in increasing Maximum Voltage sequence.

[COMMON.CHECK.PD.7#9]

11. The Tester checks that no Fixed PDO has the same voltage as any other

[COMMON.CHECK.PD.7#10]

12. The Tester checks that no Variable PDO has the same voltage range as any other

[COMMON.CHECK.PD.7#11]

13. The Tester checks that no Battery PDO has the same voltage range as any other

[COMMON.CHECK.PD.7#12]

Table 6 Max\_Current\_Field Truth Table

UUT is Source	Yes			
Src_PDO_Max_Current N in VIF	<= 3A	>3A		
Emulated eMarker	3A/5A/None	3A or None	-	5A
UUT has Captive Cable	Don't care	No	Yes	No
Expected Message PDO Max Current for check	SrcPDO_Max_curren tN <sup>2</sup>	3A	SrcPDO_Max_currentN <sub>2</sub>	

<sup>2</sup> SRCPDO\_Max\_currentN = 5A in 20V Fixed, 20V Programmable, 20V Adjustable Voltage Supply, or Variable Supply PDO if Source is operating in the EPR mode.

Table 7 Src\_PDO\_Max\_PowerN Truth Table

UUT is Source	Yes			
(Src_PDO_Max_PowerN / Src_PDO_Min_VoltageN) in VIF	<=3A	>3A		
Emulated eMarker	3A/5A/None	3A or None	-	5A
UUT has Captive Cable	Don't care	No	Yes	No
Expected Message PDO Max Power for check	SrcPDO_Max_PowerN in VIF <sup>3</sup>	(3A * Src_PDO_Min _VoltageN)	SrcPDO_Max_Po werN in VIF <sup>3</sup>	

<sup>3</sup> SrcPDO\_Max\_PowerN = 100W in 20V Battery Supply PDO if Source is operating in the EPR mode.

Table 8 PDP

	FULL CAPABILITY = Y	FULL CAPABILITY = N
Has_Invariant_PDOs = Y in VIF	PD_Power_as_Source in VIF	Calculated PDP <sup>1</sup>
Has_Invariant_PDOs = N in VIF	Calculated PDP <sup>1</sup>	Calculated PDP <sup>1</sup>

<sup>1</sup> The pseudo code below demonstrates how to obtain the Calculated PDP values.

```

PDPprev = 0;
for( PDO: fixed_PDOs ){ //for each fixed PDO 5V, 9V, 15V and 20V
    Voltage_PDO = PDO.vUnits * 50mV; // The PDO's programmed field in voltage units
    Current_PDO = PDO.iUnits * 10mA; // The PDO's programmed field in current units
    tPDP = ( Voltage_PDO * Current_PDO ) / 1000 ); // in mW or (vUnits * iUnits) /2
    Calculated_PDP = max(PDPprev , tPDP);
    PDPprev = Calculated_PDP;
}

```

#### COMMON.CHECK.PD.8 Check Request Message

Description: The Tester performs additional protocol checks to all *Request* Messages sent by the UUT.

Check Applicability: The *Request* Message

The Tester performs the following additional checks on all *Request* Messages:

#### [COMMON.CHECK.PD.8#1]

1. Field check for all types of Request Data Object
  - a. B31...28 (Object Position) is not 000b, and the value is not greater than the number of PDOs in the last *Source\_Capabilities* Message
  - b. B25 (USB Communications Capable), matches VIF field USB\_Comms\_Capable
  - c. B24 (No USB Suspend), matches VIF field No\_USB\_Suspend\_May\_Be\_Set
  - d. PD3 mode:
    1. Bit 23 (Unchunked Extended Messages Supported) matches VIF field Unchunked\_Extended\_Messages\_Supported
    2. Bit 22 (EPR Mode Capable),
      - a. Always '0b' if EPR\_Supported\_as\_Snk=0
      - b. Shall match the VIF field "EPR\_Supported\_As\_Snk" in the last request prior entering the EPR mode.
    3. Bits 21 (Reserved) set to 0b .
- PD2 mode:
 

Bits 23...20 (Reserved) set to 0000b.
2. Fixed and Variable Request Data Object
  - a. B19...10 (Operating Current) does not exceed the current offered by the referenced PDO.
  - b. B27 (Reserved) set 0b (formerly known as GiveBack). Also,

- i. If B9...0 (Maximum Operating Current) is greater than the current offered by the referenced PDO, then B26 (Capability Mismatch) is set to one.
    - ii. B19...10 (Operating Current) does not exceed B9...0 (Maximum Operating Current)
  - c. B20 (Reserved) set to 0b.
- 3. Battery Request Data Object
  - a. B19...10 (Operating Power) does not exceed the power offered by the referenced PDO.
  - b. B27 (Reserved) set to 0b (formerly known as GiveBack). Also,
    - i. If B9...0 (Maximum Operating Power) is greater than the power offered by the referenced PDO, then B26 (Capability Mismatch) is set to one.
    - ii. B19...10 (Operating Power) does not exceed B9...0 (Maximum Operating Power)
  - c. B20 (Reserved) set to 0b.
- 4. Adjustable Voltage Supply Request Data Object
  - a. B6..0 (Operating Current) is less than or equal to the Maximum Current offered by the referenced PDO.
  - b. B8..7 (Reserved) set to 00b
  - c. B20..9 (Output Voltage field):
    - i. In *Source\_Capabilities* SPR AVS APDO (B9..0) is set to 0, the valid output voltage range is [9.0-15.0]V
    - ii. Else, the valid output voltage range is [9.0-20.0]V
  - d. B27 (Reserved) set to 0b.
- 5. Programmable Power Supply Request
  - a. B6...0 (Operating Current) is less than or equal to the Maximum Current offered by the referenced PDO.
  - b. B20...9 (Output Voltage field):
    - i. Greater than or equal to the Minimum Voltage field in the PPS APDO.
    - ii. Less than or equal to the Maximum Voltage field in the PPS APDO.
  - c. B8...7 (Reserved) set to zero.
  - d. Bits 27 (Reserved) set to 0b.

#### *COMMON.CHECK.PD.9 Check Structured VDM Message*

Description: The Tester performs the protocol checks on the Structured VDM Message sent by the UUT.

Check Applicability: All Structured VDM messages.

The Tester performs the following checks on the Structured VDM Message Header (the first Data Object): **COMMON.CHECK.PD.9#1**

1. Bits 31...16 (SVID) (does not apply if a message is a response to an invalid message from the tester)
  - a. This field shall be set to 0xFF00 (PD SID) if Command (Bits 4...0) is a *Discover Identity* or *Discover SVIDs*
  - b. This field shall not be set to 0xFF00 if Command (Bits 4...0) is 16-31 (SVID specific)
2. Bit 14...13 (Structured VDM Version) are set to 00b (when the test is running in PD2 mode)



3. Bit 14...13 (Structured VDM Version) are set to 01b (when the test is running in PD3 mode)
4. Bits 12...11: (Structured VDM Version) are set to 00b in PD2 mode and 01b in PD3 mode.
5. Bits 10...8 (Object Position):
  - a. Set to 001b...111b if Command = Exit Mode
  - b. Set to 001b...110b if Command = (*Enter Mode* or *Attention*)
  - c. Set to 000b if Command = (*Discover Identity*, *Discover SVIDs* or *Discover Modes*)
6. Bit 5 (reserved) is set to 0b.
7. Bits 4...0 (Command) shall be 1-6 or 16-31

#### *COMMON.CHECK.PD.10 Check Extended Message Header*

Description: The Tester performs the protocol checks on the Message Header and Extended Message Header sent by the UUT.

Check Applicability: All PD messages except *GoodCRC*.

The Tester performs the following checks on the Message Header and Extended Message Header of all Extended Messages: *[COMMON.CHECK.PD.10#1]*

1. If both the Tester and UUT support Unchunked Extended Message (i.e. Unchunked Extended Messages Supported field set to 1b in the Tester sent Request or the *Source\_Capabilities* Message, and Unchunked Extended Messages Supported bit set to 1b (in UUT sent Request or *Source\_Capabilities* Message)):
  - a. Bit 15 Chunked bit (in Extended Message Header) = 0
  - b. For SOP'/SOP'', Bit 15 Chunked bit in Extended Message Header = 1
2. If either the Tester or the UUT supports Chunked Extended Message (i.e. either Unchunked Extended Messages Supported field set to 0b in the Tester sent the *Request* or *Source\_Capabilities* Message, or Unchunked Extended Messages Supported bit set to 0b (in UUT sent Request or the *Source\_Capabilities* Message)):
  - a. Bit 15 Chunked bit (in Extended Message Header) = 1
  - b. The payload is padded to the next 4-byte Data Object boundary with zeros.
3. Additional checks for Extended Message Header
  - a. Bits 14...11 Chunk Number = 0 if Chunked flag is also set to zero
  - b. Bit 9 Reserved = 0
  - c. Bits 8...0 Data Size
    - i. *Source\_Capabilities\_Extended* = 25
    - ii. Status
      1. For SOP shall be equal to 7
      2. For SOP' shall be equal to 2
      3. For SOP'' shall be equal to 2
    - iii. *Get\_Battery\_Status* = 1
    - iv. *Battery\_Capabilities* = 9
    - v. Manufacturer\_Info
      1. Valid Get\_Manufacturer\_Info = 4 + vendor defined null terminated string (1...22 including null)
      2. Invalid Get\_Manufacturer\_Info = 18 = (PID + VID + "Not Supported" + null)



#### COMMON.CHECK.PD.11 Check Source Capabilities Extended Message

Description: The Tester performs additional protocol checks on all

*Source\_Capabilities\_Extended* messages sent by the UUT.

Check Applicability: the *Source\_Capabilities\_Extended* Message

Perform the following checks on all *Source\_Capabilities\_Extended*:

1. For the SCEDB [COMMON.CHECK.PD.11#1]
  - a. Voltage Regulation field: Bits 0..1 (Load Step) = 00b or 01b
  - b. Voltage Regulation field: Bits 3..7 (Reserved) are 0
  - c. Compliance field: Bits 3..7 (Reserved) are 0
  - d. Touch Current field: Bits 3..7 (Reserved) are 0
  - e. Touch Temp field is 0, 1 or 2.
  - f. Source Inputs field: If Bit 0 is 0, Bit 1 is 0
  - g. Source Inputs field: Bits 3..7 (Reserved) are 0
  - h. Batteries/Battery Slots field:
    - i. Upper nibble (Number of Hot Swappable Battery Slots) <= 4 and matches VIF field Num\_Swappable\_Battery\_Slots
    - ii. Lower nibble (Number of Fixed Batteries) <= 4 and matches VIF field Num\_Fixed\_Batteries
  - i. SPR Source PDP field
    - i. if EPR\_Supported\_As\_Src is set  
Bits 0...6 (Source's rated PDP) is set to SPR PDP, which is calculated as a product of Src\_PDO\_Voltage and Src\_PDO\_Max\_Current from the maximum SPR Fixed PDO in the VIF.
    - ii. if EPR\_Supported\_As\_Src is not set  
Bits 0...6 (Source's rated PDP) is in consistent with PD\_Power\_As\_Source. If PD\_Power\_As\_Source is less than 10000mW, use the integer portion of PD\_Power\_As\_Source only to compare.
    - iii. Bit 7 (Reserved) is 0
  - j. EPR Source PDP Rating
    - i. If EPR\_Supported\_As\_Src is set
      1. Bits 0...7 EPR (Source's rated PDP) is in consistent with VIF field "PD\_Power\_As\_Source"
    - ii. If EPR\_Supported\_As\_Src is not set
      1. Bit 0...7 EPR (Source's rated PDP) is set to 0.

#### COMMON.CHECK.PD.12 Check Sink Capabilities Message

Description: The Tester performs additional protocol checks on all *Sink\_Capabilities* Messages sent by the UUT.

Check Applicability: The *Sink\_Capabilities* Message

Perform the following checks on all *Sink\_Capabilities* Messages:

[COMMON.CHECK.PD.12#1]

1. The Tester runs the following field checks on the *Sink\_Capabilities* Message:

- a. The Number of Data Objects field in the Message Header:
  - i. Is at least one.
  - ii. In PD3 mode, the check fails if the Number of Data Objects field in the Message Header is not equal to VIF field Num\_Snk\_PDOs.
  - iii. In PD2 mode, the check fails if the Number of Data Objects field in the Message Header is not equal to VIF field Num\_Snk\_PDOs minus number of PPS PDOs (count of how many Snk\_PDO\_Supply\_TypeN is set to PPS).
- b. For the first PDO, the Tester checks:
  - i. B31...30 (Fixed Supply) set to 00b.
  - ii. B29 (Dual-Role Power) set to 1b if VIF field PD\_Port\_Type is DRP, Provider/Consumer or Consumer/Provider, otherwise set to 0b.
  - iii. B28: (Higher Capability) matches VIF field Higher\_Capability\_set
  - iv. B27 - (Unconstrained\_Power) matches VIF field Unconstrained\_Power
  - v. B26 (USB Communications Capable) matches VIF field USB\_Comms\_Capable
  - vi. B25 (Dual-Role Data) set to 1b if VIF field DR\_Swap\_To\_UFP\_Supported or DR\_Swap\_To\_DFP\_Supported = YES, otherwise set to 0b if VIF field DR\_Swap\_To\_UFP\_Supported and DR\_Swap\_To\_DFP\_Supported = NO
  - vii. B24...23
    1. PD3 mode (Fast Role Swap required USB Type-C Current) matches VIF field FR\_Swap\_Reqd\_Type\_C\_Current\_As\_Initial\_Source
    2. PD2 mode: set to 00b.
  - viii. B22...20 (Reserved) set to 000b.
  - ix. B19...10 (Voltage) set to 5V.
  - x. B9...0 (Operational Current) matches VIF field SNK\_PDO\_OP\_CURRENT1
- c. For each PDO, the Tester checks:
  - i. The Tester checks consistency of Snk\_PDO\_Supply\_TypeN
  - ii. If Fixed Supply PDO:
    1. In PD2, B24...20 (Reserved) are set to zero
    2. In PD3, B22...20 (Reserved) are set to zero
    3. The Tester checks consistency of Snk\_PDO\_VoltageN
    4. The Tester checks consistency of Snk\_PDO\_Op\_CurrentN
  - iii. If Variable Supply PDO:
    1. The Tester checks consistency of Snk\_PDO\_Max\_VoltageN
    2. The Tester checks consistency of Snk\_PDO\_Min\_VoltageN
    3. The Tester checks consistency of Snk\_PDO\_Op\_CurrentN
  - iv. If Battery Supply PDO:
    1. The Tester checks consistency of Snk\_PDO\_Max\_VoltageN
    2. The Tester checks consistency of Snk\_PDO\_Min\_VoltageN
    3. The Tester checks consistency of Snk\_PDO\_Op\_PowerN

- v. If Programmable Power Supply APDO:
  - 1. B29...28 set to 00b.
  - 2. B27...25 (Reserved) set to zero.
  - 3. B16 (Reserved) set to zero.
  - 4. B7 (Reserved) set to zero.
  - 5. The Tester checks consistency of Snk\_PDO\_Max\_VoltageN
  - 6. The Tester checks consistency of Snk\_PDO\_Min\_VoltageN
  - 7. The Tester checks consistency of Snk\_PDO\_Max\_CurrentN
- d. The Tester compares all PDOs to VIF field PD\_Power\_as\_Sink and check that they meet the requirements of the Power Rules
- e. The Tester checks PDOs following the first one, are in the correct order: Fixed PDOs in increasing voltage sequence, Battery PDOs in increasing minimum voltage sequence, Variable PDOs in increasing minimum voltage sequence, and finally PPS PDOs in increasing maximum voltage sequence.
- f. The Tester checks that no Fixed PDO has the same voltage as any other.
- g. The Tester checks that no Variable PDO has the same voltage range as any other.
- h. The Tester checks that no Battery PDO has the same voltage range as any other.
- i. The Tester checks that no PPS PDO has the same voltage range as any other.

#### *COMMON.CHECK.PD.13 Check Correct Use of Rp*

**Description:** The Tester checks whether the procedures for Collision Avoidance are correctly followed.

**Check Applicability:** In PD3 mode only, and explicit contract exists.

Perform the following checks on the Atomic Message Sequences:

#### *[COMMON.CHECK.PD.13#1]*

1. The Source UUT starts an AMS:
 

When the Source UUT sends the first message to start an AMS, check that *SinkTxNG* has been asserted for at least *tSinkTx* min before the start of the first bit of the preamble.
2. An AMS initiated by the Source UUT has ended:
 

When the last message of an AMS initiated by the Source UUT has been sent, check that *SinkTxNG* is not asserted for more than *tSrcHoldsBus*. This delay is measured between the last bit of the EOP of last *GoodCRC* in the last AMS and the first bit of the preamble of the next AMS. (Refer to Source Port Policy Engine State Diagram in USB PD Specification)

**NOTE:** If this check fails, it might be because of the Source UUT asserting SinkTxOK for a very short time duration, and hence it is recommended to verify this check manually in the signal capture that the Source UUT has changed the Rp assertion from SinkTxNG to SinkTxOK between the two AMS where this failure was observed and confirm the failure.
3. The Sink UUT starts an AMS:
 

When a Sink starts an AMS, one of the following must be true:

  - a. *SinkTxOk* is asserted.
  - b. *SinkTxNG* has been asserted for less than *tSinkDelay*.

4. The following AMS are the exception for all the above steps:
  - a. *Hard Reset*
  - b. *Soft Reset*
  - c. *Alert*

#### *COMMON.CHECK.PD.14 Check Hard Reset*

Description: The Tester checks whether the basic timing for a Hard Reset is correctly followed.

Check Applicability: In PD2 or PD3 mode only, when testing a Source.

Perform the following checks on the Hard Reset Sequences: [COMMON.CHECK.PD.14#1]

1. If the Tester sees or sends a Hard Reset Signal:
  - a. Check that  $V_{BUS}$  does not go below valid range of initial voltage before *tPSHardReset* min.
  - b. Check that  $V_{BUS}$  reaches *vSafe0V* before *tPSHardReset* max plus *tSafe0V* max from leaving the valid range of initial voltage.
  - c. Check that  $V_{BUS}$  remains below *vSafe0V* for between *tSrcRecover* min and max.

Check that  $V_{BUS}$  reaches *vSafe5V* before *tSrcTurnOn* max after rising above *vSafe0V*.

**NOTE:** The Source UUT shall follow the *tSrcRecover* (EPRM\_Mode) timer when UUT is in the EPR Mode as defined in the [USB PD V1.8] EPR\_Mode Message section.

#### *COMMON.CHECK.PD.15 Check Sink\_Capabilities\_Extended Message*

Description: The Tester performs additional protocol checks on all Sink Capabilities Extended messages sent by the UUT.

Check Applicability: the *Sink\_Capabilities\_Extended* Message

Perform the following checks on all *Sink\_Capabilities\_Extended*:

For the SKEDB [COMMON.CHECK.PD.15#1]

1. SKEDB Version field (Byte 10): value is equal to 1 (Version 1.0). The check fails if the value is 0 or greater than 1.
2. Load Step field (Byte 11)
  - a. The check fails if Bits 1..0 are 10b or 11b
  - b. Bits 2..7 (Reserved) are 0.
3. Compliance field (Byte 12): Bits 3...7(Reserved) are 0
4. Touch Temp field (Byte 15): 0, 1, 2 or 3
5. Battery Info field (Byte 16):
  - i. Upper nibble (Number of Hot Swappable Battery Slots) <= 4 and matches VIF field Num\_Swappable\_Battery\_Slots
  - ii. Lower nibble (Number of Fixed Batteries) <= 4 and matches VIF field Num\_Fixed\_Batteries
6. Sink Modes field (Byte 17):
  - a. Bit 0 (PPS Charging Supported) matches VIF field Snk\_PPS\_Charging\_Supported
  - b. Bit 1 (VBUS Powered) matches VIF field Snk\_VBUS\_Powered
  - c. Bit 2 (AC Supply Powered) matches VIF field Snk\_AC\_Supply\_Powered
  - d. Bit 3 (Battery Powered) matches VIF field Snk\_Battery\_Powered
  - e. Bit 4 (Battery Essentially Unlimited) matches VIF field Snk\_Battery\_Essentially\_Unlimited. If Snk\_Battery\_Powered is not YES, this bit shall be 0

- f. Bit 5 (AVS Supported) matches VIF field Snk\_AVS\_Supported
  - g. Bits 6...7(Reserved) are 0
- 7. Sink Minimum PDP field (Byte 18):
  - i. Bits 0...6 (Sink's minimum PDP) is less than or equal to 100.
  - ii. Bits 0...6 (Sink's minimum PDP) is less than or equal Sink Operational PDP
  - iii. Bits 0...6 (Sink's minimum PDP) is equal to VIF field Snk\_SPR\_Minimum\_PDP
  - iv. Bit 7(Reserved) is 0
- 8. Sink Operational PDP field (Byte 19):
  - i. Bits 0...6 (Sink's minimum PDP) is less than or equal to 100.
  - ii. Bits 0...6 (Sink's minimum PDP) is equal to VIF field Snk\_SPR\_Operational\_PDP
  - iii. Bit 7(Reserved) is 0
- 9. Sink Maximum PDP field (Byte 20):
  - i. Bits 0...6 (Sink's minimum PDP) is less than or equal to 100.
  - ii. Bits 0...6 (Sink's minimum PDP) is equal or greater than Sink Operational PDP
  - iii. Bits 0...6 (Sink's minimum PDP) is equal to VIF field Snk\_SPR\_Maximum\_PDP
  - iv. Bit 7(Reserved) is 0
- 10. EPR Sink Minimum PDP field (Byte 21):
  - i. if EPR\_Support\_As\_Snk is not set
    - 1. Bits 0...7 (EPR Sink's minimum PDP) is set to 0.
  - ii. if EPR\_Support\_As\_Snk is set
    - 1. Bits 0...7 (EPR Sink's minimum PDP) is less than or equal to EPR Sink operational PDP.
    - 2. Bits 0...7 is less than or equal to 240
    - 3. Bits 0...7 (EPR Sink's minimum PDP) is less than or equal to VIF field Snk\_EPR\_Minimum\_PDP
- 11. EPR Sink Operational PDP field (Byte 22):
  - i. if EPR\_Support\_As\_Snk is not set
    - 1. Bits 0...7 (EPR Sink's operational PDP) is set to 0.
    - 2. Bits 0...7 (EPR Sink's operational PDP) is equal to VIF field Snk\_EPR\_Operational\_PDP
- 12. EPR Sink Maximum PDP field (Byte 23):
  - i. if EPR\_Support\_As\_Snk is not set
    - 1. Bits 0...7 (EPR Sink's maximum PDP) is set to 0.
  - ii. if EPR\_Support\_As\_Snk is set
    - 1. Bits 0...7 (EPR Sink's maximum PDP) is greater than or equal to EPR Sink operational PDP field and
    - 2. Bits 0...7 is less than or equal to 240
    - 2. Bits 0...7(EPR\_Snk's maximum PDP) is equal to VIF field Snk\_EPR\_Masimum\_PDP

### 3.1.2 PD3 Mode

The common procedures in this subsection are applicable when the test is running in PD3 mode.

#### COMMON.CHECK.PD3.1 Check *EPR\_Request* Message

Description: The Tester performs additional protocol checks to all *EPR\_Request* Messages sent by the UUT.

Check Applicability: the *EPR\_Request* Message

The Tester performs the following additional checks on all *EPR\_Request* Messages:

#### [COMMON.CHECK.PD3.1#1]

1. Field check for all types of Request Data Object
  - a. B31...28 (Object Position) is not 000b, and the value is not greater than the number of PDOs in the last the *EPR\_Source\_Capabilities* Message
  - b. B25 (USB Communications Capable), matches VIF field USB\_Comms\_Capable
  - c. B24 (No USB Suspend), matches VIF field No\_USB\_Suspend\_May\_Be\_Set
  - d. Bit 23 (Unchunked Extended Messages Supported) matches VIF field Unchunked\_Extended\_Messages\_Supported if the test is running in PD3 mode.
  - e. Bits 22 (EPR Mode Capable) set to 1b.
  - f. Bits 21 (Reserved) set to 0b.
2. Fixed and Variable Request Data Object
  - a. B19...10 (Operating Current) does not exceed the current offered by the referenced PDO.
  - b. B27 (Reserved) is set 0b (Formerly known as GiveBack). Also,
    - i. If B9...0 (Maximum Operating Current) is greater than the current offered by the referenced PDO, then B26 (Capability Mismatch) is set to one.
    - ii. B19...10 (Operating Current) does not exceed B9...0 (Maximum Operating Current)
  - c. B20 (Reserved) set to 0b.
3. Battery Request Data Object
  - a. B19...10 (Operating Power) does not exceed the power offered by the referenced PDO.
  - b. B27 (Reserved) is set to 0b (Formerly known as GiveBack). Also,
    - ii. If B9...0 (Maximum Operating Power) is greater than the power offered by the referenced PDO, then B26 (Capability Mismatch) is set to one.
    - i. B19...10 (Operating Power) does not exceed B9...0 (Maximum Operating Power)
  - c. B20 (Reserved) set to 0b.
4. SPR Adjustable Voltage Supply RDO.
  - a. B6..0 (Operating Current) is less than or equal to the Maximum Current by the referenced PDO.
  - b. B8..7 (Reserved) set to 00b.
  - c. B20..9 (Output Voltage field):
    - i. In *EPR\_Source\_Capabilities* SPR AVS APDO (B9..0) is set to 0, the valid output voltage range is [9.0-15.0]V

- ii. Else, the valid output voltage range is [9.0-20.0]V.
- d. B27 (Reserved) set to 0b.
- 5. SPR Programmable Power Supply RDO
  - a. B6...0 (Operating Current) is less than or equal to the Maximum Current offered by the referenced PDO.
  - b. B20...9 (Output Voltage field):
    - i. Greater than or equal to the Minimum Voltage field in the PPS APDO.
    - ii. Less than or equal to the Maximum Voltage field in the PPS APDO.
  - c. B8...7 (Reserved) set to zero.
  - d. Bits 27 (Reserved) set to 0b.
- 6. EPR Adjustable Voltage Supply Request RDO
  - a. B6...0 (Operating Current) is less than or equal to the Source PDP/Output Voltage rounded down to the nearest 50mA.
  - b. B20...9 (Output Voltage field):
    - i. Greater than or equal to the Minimum Voltage field in the AVS APDO.
    - ii. Less than or equal to the Maximum Voltage field in the AVS APDO.
  - c. B8...7 (Reserved) set to zero.
  - d. Bits 27 (Reserved) set to 0b.
- 7. Tester checks that the PDO(PDO2) in the *EPR\_Request* Message exactly matches the PDO in the latest the *EPR\_Source\_Capabilities* Message pointed to by the *Object Position* field in the RDO.

#### *COMMON.CHECK.PD3.2 Check EPR\_Mode Message*

Description: The Tester performs additional protocol checks to all *EPR\_Mode* Messages sent by the UUT.

Check Applicability: the *EPR\_Mode* Message

The Tester performs the following additional checks on all EPR Mode messages:

#### *[COMMON.CHECK.PD3.2#1]*

- 1. Field check for EPR Mode Data Object
  - a. Bit 31...24 (Action field) shall not be equal to 0 and shall be less than 0x06
  - b. Bit 23...16 (Data field):
    - i. Shall be set to 0 if the action field is “Enter Acknowledged” (0x02)
    - ii. Shall be set to 0 if the action field is “Enter Succeeded” (0x03)
    - iii. Shall be set to 0 if the action field is “Exit” (0x05)
    - iv. Shall be consistent with the “Snk\_EPR\_Operational\_PDP” VIF field if action field is “Enter” (0x01)
    - v. Shall be less than 0x06 if the action field is “Enter Failed” (0x04)
  - c. Bit 15...0 Shall be set to 0.

### COMMON.CHECK.PD3.3 Check EPR\_Source\_Capabilities Message

**Description:** The Tester performs additional protocol checks on all *EPR\_Source\_Capabilities* Messages sent by the UUT.

Depending on the UUT's internal power policies, the UUT may offer power that is lower than in the VIF PD\_Power\_As\_Source field and as a result the PDOs will not match PDOs specified in the VIF. For those types of UUTs the Tester would run reduced set of checks. On the other hand, if the UUT offers full power all the time despite of internal or external conditions, the Tester would run the fullset of checks in this test. To differentiate between two types of UUTs, the Tester uses VIF Parameter Has\_Invariant\_PDOs:

- If VIF Parameter Has\_Invariant\_PDOs is set to Y – run all checks.
- If VIF Parameter Has\_Invariant\_PDOs is set to N – run all checks except those marked 'FULL CAPABILITY ONLY'

**Check Applicability:** the *EPR\_Source\_Capabilities* Message

**NOTE:** This check is run for TEST.PD.EPR.SRC and all test procedures where *EPR\_Source\_Capabilities* are sent by the UUT.

Perform the following checks on all *EPR\_Source\_Capabilities* Messages:

1. Check if the VIF field EPR\_Supported\_as\_Src is set to yes.  
[COMMON.CHECK.PD3.3#1]
2. The Tester checks Extended field = 1 in the Message Header.  
[COMMON.CHECK.PD3.3#8]
3. FULL CAPABILITY ONLY: The Tester checks Data size in the extended header: (SPR PDOs(7) + EPR PDO entries)\*4. The number of EPR PDOs is determined by counting the total number of entries in SrcPdoList and subtracting Num\_Src\_PDOs VIF Parameter. [COMMON.CHECK.PD3.3#9]
4. For the first PDO, the Tester check the consistency [COMMON.CHECK.PD3.3#2]
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 1b if VIF field PD\_Port\_Type is set to Consumer/Provider, Provider/Consumer or DRP.
  - c. B28 (USB Suspend Supported) set to 1b if VIF field USB\_Suspend\_May\_Be\_Cleared = NO
  - d. B27 (Unconstrained Power) matches VIF field Unconstrained\_Power
  - e. B26 (USB Communications Capable) matches VIF field USB\_Comms\_Capable
  - f. B25 (Dual-Role Data) set to 1b if VIF field DR\_Swap\_To\_UFP\_Supported or DR\_Swap\_To\_DFP\_Supported = YES
  - g. Bit 24 (Unchunked Extended Messages Supported) matches VIF field Unchunked\_Extended\_Messages\_Supported if the test is running in the PD3 mode.
  - h. Bit 23 (EPR Mode Capable) matches VIF field EPR\_Supported\_as\_Src if the test is running in the PD3 mode.
  - i. Bits 22 (Reserved) set to 00b if the test is running in the PD3 mode.
  - j. B21...20 (Peak Current) matches VIF field Src\_PDO\_Peak\_Current1
  - k. B19...10 (Voltage) set to 5V.



1. B9...0 Max\_current\_field
    1. For the UUT as a Sink the current shall be Src\_PDO\_Max\_Current1
    2. FULL CAPABILITY ONLY: Check the current shall be Src\_PDO\_Max\_Current1
- NOTE:** For the truth table of **Max\_Current\_Field**, which is **Src\_PDO\_Max\_current1**, see [Table 6](#) Max\_Current\_Field Truth Table*
5. If it is not the first Fixed Supply PDO, check that B29...22 is set to zero [\[COMMON.CHECK.PD3.3#3\]](#)
  6. If it is a Programmable Power Supply APDO, check that B26...25, B16 and B7 are set to zero [\[COMMON.CHECK.PD3.3#4\]](#)
  7. Check that the UUT followed all the PDO specific checks defined in COMMON.CHECK.PD.7 for all the SPR PDO in PDO position between 1 and 7. For the following cases, check the condition below instead. [\[COMMON.CHECK.PD3.3#10\]](#)
    - a. Maximum Current = 5A in SPR 20V Fixed, 20V Programmable, 20V Adjustable Voltage Supply PDO.
    - b. Operational current field in SPR Variable Supply PDO = 5A.
    - c. Allowable Power in Battery Supply PDO = 100W.
  8. The Tester compares all the EPR PDOs (position 8 to 11) to PDP ([Table 8](#)) and check that they meet the requirements of the Power Rules. [\[COMMON.CHECK.PD3.3#5\]](#)
    - a. Fixed PDO
      1. Check that maximum current is RoundUp (PD\_Power\_as\_Source/Voltage) or RoundDown (PD\_Power\_as\_Source/Voltage) to the nearest 10mA.
      2. Check that all the normative voltages are supported by UUT based on the PDP as per the Table “EPR Source Capabilities based on the Port Maximum PDP and using as EPR Capable Cable” in the PD3 specification.
      3. Check that the Voltage is either 28V, 36V or 48V.
    - b. EPR AVS APDO
      1. Check that maximum current is RoundUp (PD\_Power\_as\_Source/Voltage) or RoundDown (PD\_Power\_as\_Source/Voltage) to the nearest 10mA.
      2. Check that all the normative voltages are supported by the UUT based on the PDP as per the Table “EPR Source Capabilities based on the Port Maximum PDP and using an EPR Capable Cable” in the PD3 specification.
      3. Check that the Minimum and Maximum Voltages are per Table “EPR Adjustable Voltage Supply (AVS) Voltage Ranges” in the PD3 specification.
      4. Check that the Peak Current field matches with the Src\_PDO\_Peak\_Current VIF field.
    - c. Check that the UUT does not support any optional voltages in the EPR PDO position (PDO index from 8 to 11)

- d. Check that the UUT supports only Fixed PDO and EPR AVS APDO in the EPR PDO position (PDO index from 8 to 11)
- e. If the UUT supports less than 7 SPR PDO's check that the unused SPR PDO's are zero filled in the SPR PDO positions (PDO index from 1 to 7)
- f. Check that the UUT supports only one EPR AVS APDO in the last EPR PDO position.
- g. The Tester checks PDOs following the first one, are in the correct order: Fixed PDOs in the increasing Voltage sequence.
- h. The Tester checks that no Fixed PDO has the same voltage as any other.
9. FULL CAPABILITY ONLY: For each EPR PDO (position 8 to 11), the Tester checks VIF consistency of: [\[COMMON.CHECK.PD3.3#6\]](#)
  - a. If EPR Adjustable Voltage Supply APDO:
    1. Src\_PDO\_MinVoltageN
    2. Src\_PDO\_MaxVoltageN
  - b. If Fixed Supply PDO:
    1. Src\_PDO\_VoltageN
    2. Src\_PDO\_Peak\_CurrentN
    3. Src\_PDO\_Max\_CurrentN

#### [COMMON.CHECK.PD3.4 Check EPR\\_Sink\\_Capabilities Message](#)

**Description:** The Tester performs additional protocol checks on all EPR *Sink\_Capabilities* Messages sent by the UUT.

**Check Applicability:** the *EPR\_Sink\_Capabilities* Message

Perform the following checks on all *EPR\_Sink\_Capabilities* Messages:

1. Check if the VIF field EPR\_Supported\_as\_Snk is set to yes and *Get\_EPR\_Snk\_Capabilities* have been sent by the tester. [\[COMMON.CHECK.PD3.4#1\]](#)
2. The Tester checks Extended field = 1 in the Message Header. [\[COMMON.CHECK.PD3.4#2\]](#)
3. The Tester checks data size in the extended header: ( 7 \* SPR PDOs + X \* EPR PDOs) \*4. Where X is the number of EPR PDOs determined by counting the total number of entries in SnkPdoList minus Num\_Snk\_PDOs VIF Parameter. [\[COMMON.CHECK.PD3.4#3\]](#)
4. For the first PDO, the Tester checks: [\[COMMON.CHECK.PD3.4#4\]](#)
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 1b if VIF field PD\_Port\_Type is DRP, Provider/Consumer or Consumer/Provider, otherwise set to 0b.
  - c. B28: (Higher Capability) matches VIF field Higher\_Capability\_set
  - d. B27 - (Unconstrained\_Power) matches VIF field Unconstrained\_Power
  - e. B26 (USB Communications Capable) matches VIF field USB\_Comms\_Capable
  - f. B25 (Dual-Role Data) set to 1b if VIF field DR\_Swap\_To\_UFP\_Supported or DR\_Swap\_To\_DFP\_Supported = YES, otherwise set to 0b if VIF field DR\_Swap\_To\_UFP\_Supported and DR\_Swap\_To\_DFP\_Supported = NO
  - g. B24...23 PD3 mode (Fast Role Swap required USB Type-C Current) matches VIF field FR\_Swap\_Reqd\_Type\_C\_Current\_As\_Initial\_Source

- h. B22...20 (Reserved) set to 000b.
  - i. B19...10 (Voltage) set to 5V.
  - j. B9...0 (Operational Current) matches VIF field SNK\_PDO\_OP\_CURRENT1
5. Check that the UUT followed all PDO specific checks defined in COMMON.CHECK.PD.12 for all SPR PDO in PDO position between 1 and 7.  
[COMMON.CHECK.PD3.4#5]
  6. The Tester checks the EPR PDOs (position 8 to 11) meet the requirements of the Sink power rules [COMMON.CHECK.PD3.4#6]
    - a. Check that the UUT supports only Fixed PDO or EPR AVS APDO in the EPR PDO position (PDO index from 8 to 11).
    - b. If the UUT supports less than 7 SPR PDO's check that the unused SPR PDO's are zero filled in the SPR PDO positions (PDO index from 1 to 7).
    - c. Check that the UUT supports no more than one EPR AVS APDO.
    - d. The Tester checks PDOs following the first one, are in the correct order: Fixed PDOs are in the increasing Voltage sequence.
    - e. The Tester checks that Fixed PDOs do not have the same voltage.

#### COMMON.CHECK.PD3.5 Check *tEPRSourceCableDiscovery*

Description: Before entering the EPR mode the Tester verifies that the Discover Identity SOP' information has been obtained from a cable.

Check Applicability: All *PS\_RDY* Messages prior entering the EPR mode.

If the VIF field EPR\_Supported\_As\_Src is Yes, the UUT is a provider, and after either

- a Port Partner is attached,
- Hard Reset,
- Power Role Swap or
- Fast Role Swap

the Tester checks an SOP' is sent by the UUT either: [COMMON.CHECK.PD3.5#1]

- a. Before entering the first explicit contract or
- b. Within *tEPRSourceCableDiscovery* of receiving the *GoodCRC* Message acknowledging the *PS\_RDY* Message of the first Explicit Contract negotiation.

#### COMMON.CHECK.PD3.6 The Ping Message

Description: The Tester is making sure that the Ping Message is not being used.

Check Applicability: In the PD3 mode only.

1. If the Tester receives the Ping Message, the Tester fails the check, and the test stops here.

## 3.2 Bring-up Procedures.

### 3.2.1 PD2 and PD3 Modes

The common checks in this subsection are applicable when the test is running in PD2 and PD3 modes.

#### COMMON.PROC.BU.1 Bring-up Source UUT

1. The test starts in a disconnected state.
2. If VIF field *Captive\_Cable* = YES, the Tester applies Rd. Otherwise (VIF field *Captive\_Cable* = No), the Tester applies Rd and Ra.
3. The check fails if the first bit of a *Source\_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after V<sub>BUS</sub> present (*vSafe5V*).  
[COMMON.PROC.BU.1#1]
4. The Tester responds with a *GoodCRC*
5. The Tester sends a *Request* Message for 5V at 100mA.
  - a. B30...B28 (Object Position) = 001b
  - b. B27 (Reserved) = 0b
  - c. B26 (Capability Mismatch) = 0b
  - d. B25 (USB Communication Capable) = 0b
  - e. B24 (No USB Suspend) = 1b
  - f. B23 (PD3, Unchunked Extended Messages Supported) = 0b
  - g. B22 (EPR Mode Capable) to 0b, unless it is mentioned in the test procedure
  - h. B19...10 (Operating Current) = 100mA
  - i. B9...0 (Maximum Operating Current) = 100mA
6. The check fails if: [COMMON.PROC.BU.1#2]
  - a. The UUT retries the *Source\_Capabilities* Message
  - b. The UUT does not respond with an *Accept* or *Wait* Message and the test stops here upon failure.
7. If UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after *nCtsWaitCount* retries and the test stops here. [COMMON.PROC.BU.1#4]
8. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [COMMON.PROC.BU.1#3]
9. An explicit contract is now established.
10. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT.

#### COMMON.PROC.BU.2 Bring-up Sink UUT

1. The test starts in a disconnected state.
2. The tester applies Rp default.
3. If the UUT attachment is detected, the Tester applies VCONN (if Ra is detected) and *vSafe5V* on V<sub>BUS</sub>.
4. The Tester waits for 50 ms.
5. The Tester transmits *Source\_Capabilities* Message with single PDO:
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 1b.
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B27 (Unconstrained Power) set to 1b.
  - e. B26 (USB Communications Capable) set to 0b.
  - f. B25 (Dual-Role Data) set to 0b.

- g. B24 (PD3, Unchunked Extended Messages Supported) set to 0b.
  - h. B23 (EPR Mode Capable) to 0b, unless it is mentioned in the test procedure.
  - i. B21...20 (Peak Current) set to 00b.
  - j. B19...10 (Voltage) set to 5V.
  - k. B9...0 (Maximum Current) set to 500mA.
6. Repeat Step-5 if the Tester does not receive a *GoodCRC* from the UUT in response to *Source\_Capabilities* Message. This requires a delay of *tTypeCSendSourceCap* (min to max). This sequence is repeated at least 50 times.
  7. The check fails if the UUT does not respond with a *Request* Message.  
[COMMON.PROC.BU.2#1]
  8. The Tester responds with an *Accept* Message to the *Request* Message.
  9. When the  $V_{BUS}$  voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at *tSrcTransReq* max after the reception of the last bit of the *GoodCRC* Message EOP corresponding to the *Accept* Message.
  10. An explicit contract is now established.
  11. The Tester presents *SinkTxOk* if the test is in PD3 mode. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT.

#### COMMON.PROC.BU.3 Bring-up Cable Plug UUT

1. The test starts in a disconnected state.
2. The Tester applies  $R_p$  to each of the CC1 and CC2 pins at one cable end. The other end of the cable is left unconnected.
3. The Tester checks that it detects  $R_a$  on one pin (from now on called the VCONN pin) and no connection on the other pin (from now on called the CC pin).  
[COMMON.PROC.BU.3#1]. For any other combination the test ends here.
4. The Tester performs the following:
  - a. keeps  $R_p$  connected to the CC pin and also connects  $R_d$  to the CC pin,
  - b. waits *tCCDebounce* nominal,
  - c. disconnects  $R_p$  from the VCONN pin and applies VCONN (at the voltage specified in the Test Condition) to the VCONN pin,
  - d. connects *vSafe5V* on  $V_{BUS}$ .
5. The Tester waits for *tVCONNStable* max.

**NOTE:** The Tester uses SOP' messages in Cable Plug tests except where otherwise specified.

#### COMMON.PROC.BU.4 Bring-up Source UUT for PPS Test

1. The test starts in a disconnected state.
2. The Tester applies  $R_d$ .
3. The check fails if the first bit of a *Source\_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after  $\bar{V}_{BUS}$  present (*vSafe5V*).  
[COMMON.PROC.BU.4#1]
4. If there is no PPS APDO in the *Source\_Capabilities*, the test stops here.
5. The Tester responds with a *Request* Message for Programmable Power Supply APDO 5V 1A.
  - a. B26 (Capability Mismatch) = 0b
  - b. B25 (USB Communication Capable) = 0b

- c. B24 (No USB Suspend) = 1b
  - d. B23 (PD3, Unchunked Extended Messages Supported) = 0b
  - e. B19...9 (Output Voltage) = 5V
  - f. B6...0 (Operating Current) = 1A
6. The check fails if either *Accept* or *Wait* Message is not received, and the test stops here. [COMMON.PROC.BU.4#2]
  7. If the UUT responds with the *Wait* Message, the Tester resends the *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until the *Accept* Message received. The check fails if the UUT responds with the *Wait* Message after *nCtsWaitCount* retries, and the test stops here. [COMMON.PROC.BU.4#4]
  8. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted the *Accept* Message. [COMMON.PROC.BU.4#3]
  9. An explicit contract is now established.
  10. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT.

#### COMMON.PROC.BU.5 Bring-up Sink UUT for PPS Test

1. The test starts in a disconnected state.
2. The tester applies Rp 3A.
3. If the UUT attachment is detected, the Tester applies VCONN (if Ra is detected) and *vSafe5V* on V<sub>BUS</sub>.
4. The Tester transmits the *Source\_Capabilities* Message with five PDOs:
  - a. First Fixed Supply PDO
    - i. B31...30 (Fixed Supply) set to 00b.
    - ii. B29 (Dual-Role Power) set to 1b.
    - iii. B28 (USB Suspend Supported) set to 0b.
    - iv. B27 (Unconstrained Power) set to 1b.
    - v. B26 (USB Communications Capable) set to 0b.
    - vi. B25 (Dual-Role Data) set to 0b.
    - vii. B24 (PD3, Unchunked Extended Messages Supported) set to 0b.
    - viii. B21...20 (Peak Current) set to 00b.
    - ix. B19...10 (Voltage) set to 5V.
    - x. B9...0 (Maximum Current) set to 3A.
  - b. Following 3 Fixed PDOs:
    - i. B31...30 (Fixed Supply) set to 00b.
    - ii. B29 through B20 set to 0b.
    - iii. B19...10 (Voltage) set to 9V, 15V and 20V.
    - iv. B9...0 (Maximum Current) set to 3A
  - c. APDO PPS
    - i. B31...30 (Augmented PDO) set to 11b.
    - ii. B29...28 00b – Programmable Power Supply
    - iii. B27 (PPS Power Limited) set to 0b.
    - iv. B24...17 (Maximum Voltage) set to 21V.
    - v. B15...8 (Minimum Voltage) set to 5.0V
    - vi. B6...0 (Maximum Current) set to 3A.



5. Repeat Step-4 if the Tester does not receive a *GoodCRC* Message from the UUT in response to the *Source\_Capabilities* Message. This requires a delay of *tTypeCSendSourceCap* (min to max). This sequence is repeated at least 50 times.
6. The check fails if a *Request* Message is not received. [COMMON.PROC.BU.5#1]
7. The Tester responds with an *Accept* Message to the *Request* Message.
8. When the  $V_{BUS}$  voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at *tSrcTransReq* max after the reception of the last bit of the *GoodCRC* Message EOP corresponding to the *Accept* Message.
9. An explicit contract is now established.
10. The Tester presents *SinkTxOK*.
11. If the Sink doesn't request the PPS APDO in 5s from the time Tester presented *SinkTxOK*, the test stops here. If the *Request* Message is received, the Tester responds with an *Accept* Message.
12. When the  $V_{BUS}$  voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at *tSrcTransReq* max after the reception of the last bit of the *GoodCRC* Message EOP corresponding to the *Accept* Message.
13. The Tester presents *SinkTxOK*. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT.

#### COMMON.PROC.BU.6 Bring-up Sink UUT for Captive Cable Test

1. The test starts in a disconnected state.
2. The Tester applies  $R_p$  default to each of the CC1 and CC2 pins at the captive cable end.
3. Tester waits for  $R_d$  to appear on at least one pin. If this does not occur within  $tDRP$ , the check fails. [COMMON.PROC.BU.6#1].
4. The Tester checks that it detects  $R_a$  on one pin (from now on called the  $V_{CONN}$  pin) and  $R_d$  on the other pin (from now on called the CC pin). [COMMON.PROC.BU.6#2]. For any other combination the test ends here.
5. The Tester performs the following:
  - a. keeps  $R_p$  connected to the CC pin,
  - b. waits *tCCDebounce*,
  - c. disconnects  $R_p$  from the  $V_{CONN}$  pin and applies  $V_{CONN}$  (at the voltage specified in the Test Condition) to the  $V_{CONN}$  pin,
  - d. connects *vSafe5V* on  $V_{BUS}$ .
6. The Tester waits for 50 ms.
7. The Tester transmits *Source\_Capabilities* Message with single PDO:
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 1b.
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B27 (Unconstrained Power) set to 1b.
  - e. B26 (USB Communications Capable) set to 0b.
  - f. B25 (Dual-Role Data) set to 0b.
  - g. B24 (PD3, Unchunked Extended Messages Supported) set to 0b.
  - h. B21...20 (Peak Current) set to 00b.
  - i. B19...10 (Voltage) set to 5V.
  - j. B9...0 (Maximum Current) set to 500mA.

8. Repeat Step-5 if the Tester does not receive a *GoodCRC* from the UUT in response to *Source\_Capabilities* Message. This requires a delay of *tTypeCSendSourceCap* (min to max). This sequence is repeated at least 50 times.
  9. The check fails if the UUT does not respond with a *Request* Message.  
[COMMON.PROC.BU.6#3]
  10. The Tester responds with an *Accept* Message to the *Request* Message.
  11. When the  $V_{BUS}$  voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at *tSrcTransReq* max after the reception of the last bit of the *GoodCRC* Message EOP corresponding to the *Accept* Message.
  12. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT.
  13. The Tester sends a BIST request to the UUT using SOP, specifying BIST Test Data.
- NOTE:** The Tester uses SOP' messages in Captive Cable Plug tests except where otherwise specified.

#### COMMON.PROC.BU.7 Bring-up Consumer-Provider as Source UUT

1. The test starts in a disconnected state.
2. The tester applies  $R_p$ .
3. If the UUT attachment is detected, the Tester applies VCONN (if  $R_a$  is detected) and *vSafe5V* on  $V_{BUS}$ .
4. The Tester waits for 50 ms.
5. The Tester transmits *Source\_Capabilities* Message with single PDO:
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 1b.
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B27 (Unconstrained Power) set to 1b.
  - e. B26 (USB Communications Capable) set to 0b.
  - f. B25 (Dual-Role Data) set to 0b.
  - g. B24 (PD3, Unchunked Extended Messages Supported) set to 0b.
  - h. B21...20 (Peak Current) set to 00b.
  - i. B19...10 (Voltage) set to 5V.
  - j. B9...0 (Maximum Current) set to 500mA.
6. Repeat Step-5 if the Tester does not receive a *GoodCRC* from the UUT in response to *Source\_Capabilities* Message. This requires a delay of *tTypeCSendSourceCap* (min to max). This sequence is repeated at least 50 times.
7. The check fails if the UUT does not respond with a *Request* Message.  
[COMMON.PROC.BU.7#1]
8. The Tester responds with an *Accept* Message to the *Request* Message.
9. When the  $V_{BUS}$  voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at *tSrcTransReq* max after the reception of the last bit of the *GoodCRC* Message EOP corresponding to the *Accept* Message.
10. An explicit contract is now established.
11. The Tester presents *SinkTxOk* if the test is in PD3 mode. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT. If during this period, the UUT requests a *PR\_Swap*, the Tester sends an *Accept*, and then skips to step 14.



12. The Tester sends a *PR\_Swap* Message.
13. The Tester checks the response. [COMMON.PROC.BU.7#2]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *PR\_Swap* every  $tPRSwapWait + tCtsWait$  for  $nCtsWaitCount$  times. Once the Tester has resent *PR\_Swap* for  $nCtsWaitCount$  times, the check fails, and the test ends here.
  - b. The check fails and the test ends here if the UUT responds with a *Reject* Message.
  - c. The check fails and the test ends here if the UUT response with a *Not\_Supported* Message.
14. At  $tSrcTransition$  max (this delay is from the last bit of *GoodCRC* sent in response to the *Accept*), the Tester drives the  $V_{BUS}$  voltage to  $vSafe0V$  within  $tSrcSwapStdbby$  max.
15. The Tester sends a *PS\_RDY* at the deadline limit of  $tPSSourceOff$  min, the delay is from the time the last bit of the EOP of the *GoodCRC* corresponding to the *Accept* Message.
16. The tester checks that UUT asserts  $R_p$  before the UUT has applied  $vSafe5V$  [COMMON.PROC.BU.7#3]
17. The Tester checks that the UUT sends *PS\_RDY* only after the UUT has applied  $vSafe5V$ . [COMMON.PROC.BU.7#4]
18. The Tester checks that the *PS\_RDY* from the UUT is within  $tPSSourceOn$  min. [COMMON.PROC.BU.7#5] This delay is from the *GoodCRC* sent by the UUT in response to the *PS\_RDY* sent by the Tester.
19. The Tester checks that the UUT sends a *Source\_Capabilities* Message after  $tSwapSourceStart$  and before  $tFirstSourceCap$  max. [COMMON.PROC.BU.7#6].
20. The Tester responds with a *Request* Message for 5V at 100mA.
  - a. B30 (Object Position) = 001b
  - b. B27 (Reserved) = 0b
  - c. B26 (Capability Mismatch) = 0b
  - d. B25 (USB Communication Capable) = 0b
  - e. B24 (No USB Suspend) = 1b
  - f. B23 (PD3, Unchunked Extended Messages Supported) = 0b
  - g. B19...10 (Operating Current) = 100mA
  - h. B9...0 (Maximum Operating Current) = 100mA
21. The check fails if the UUT does not respond with either *Accept* or *Wait* Message, and the test stops here. [COMMON.PROC.BU.7#7]
22. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every  $tSinkRequest$  for  $nCtsWaitCount$  times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* message after  $nCtsWaitCount$  retries, and the test stops here. [COMMON.PROC.BU.7#10]
23. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within  $tSrcTransReq$  from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [COMMON.PROC.BU.7#8]
24. An explicit contract is now established.
25. The Tester waits  $tCtsAfterLastPoint$  to respond to messages from the UUT.
26. If the UUT is not a *Vconn* Source, then the Tester sends a *VCONN\_Swap* message and completes the *VCONN\_Swap* using common procedure COMMON.PROC.PD.17. [COMMON.PROC.BU.7#9]

#### COMMON.PROC.BU.8 Bring-up Provider-Consumer as Sink UUT

1. The test starts in a disconnected state.
2. If VIF field *Captive\_Cable* = YES, the Tester applies *Rd*. Otherwise (VIF field *Captive\_Cable* = No), the Tester applies *Rd* and *Ra*.
3. The check fails if the first bit of a *Source\_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after *V<sub>BUS</sub>* present (*vSafe5V*).  
[COMMON.PROC.BU.8#1]
4. The Tester responds with a *Request* Message for 5V at 100mA.
  - a. B30...B28 (Object Position) = 001b
  - b. B27 (Reserved) = 0b
  - c. B26 (Capability Mismatch) = 0b
  - d. B25 (USB Communication Capable) = 0b
  - e. B24 (No USB Suspend) = 1b
  - f. B23 (PD3, Unchunked Extended Messages Supported) = 0b
  - g. B19...10 (Operating Current) = 100mA
  - h. B9...0 (Maximum Operating Current) = 100mA
5. The check fails if the UUT does not respond with either *Accept* or *Wait* Message, and the test stops here. [COMMON.PROC.BU.8#2]
6. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after *nCtsWaitCount* retries, and the test stops here. [COMMON.PROC.BU.8#9]
7. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [COMMON.PROC.BU.8#3]
8. An explicit contract is now established.
9. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT. If during this period, the UUT requests a *PR\_Swap*, the Tester sends an *Accept*, and then skips to step 12.
10. The Tester sends a *PR\_Swap* Message.
11. The Tester checks the response. [COMMON.PROC.BU.8#4]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *PR\_Swap* every *tPRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. Once the Tester has resent *PR\_Swap* for *nCtsWaitCount* times, the check fails, and the test ends here.
  - b. If the UUT responds with *Reject* or *Not\_Supported*, the check fails, and the test ends here.
  - c. If the UUT responds with an *Accept* message, proceed to Step 12.
12. The Tester draws no more than *iSwapStandby* within *tSrcTransition* min after the last *GoodCRC*.
13. The Tester checks that the UUT asserts *Rd* and sends a *PS\_RDY* Message after the UUT driving *V<sub>BUS</sub>* voltage to *vSafe0V*. [COMMON.PROC.BU.8#5]
14. The Tester checks that the UUT sends *PS\_RDY* within *tPSSourceOff* min after the last bit of *GoodCRC* was sent in response to the received *Accept*. [COMMON.PROC.BU.8#6]

15. The Tester presents Rp default on the CC pin and *vSafe5V* on V<sub>BUS</sub> and then it sends a *PS\_RDY* Message, both within *tNewSRC* max of the last *GoodCRC* sent by the UUT.
16. The Tester transmits *Source\_Capabilities* Message with single PDO (5V @ 500mA, B29 Dual-Role Power set to 1b). The Tester checks that the UUT sends a *Request* Message. [COMMON.PROC.BU.8#7]
17. The Tester uses the common procedure COMMON.PROC.PD.10 to establish a PD contract. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT.
18. If the Tester is not a V<sub>CONN</sub> Source, then the Tester sends a *VCONN\_Swap* message and completes the *VCONN\_Swap* using common procedure COMMON.PROC.PD.17. [COMMON.PROC.BU.8#8]

#### COMMON.PROC.BU.9 Bring-up Consumer-Provider as Source UUT PPS

1. The test starts in a disconnected state.
2. The tester applies Rp default.
3. If the UUT attachment is detected, the Tester applies VCONN (if Ra is detected) and *vSafe5V* on V<sub>BUS</sub>.
4. The Tester waits for 50 ms.
5. The Tester transmits *Source\_Capabilities* Message with single PDO:
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 1b.
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B27 (Unconstrained Power) set to 1b.
  - e. B26 (USB Communications Capable) set to 0b.
  - f. B25 (Dual-Role Data) set to 0b.
  - g. B24 (PD3, Unchunked Extended Messages Supported) set to 0b.
  - h. B21...20 (Peak Current) set to 00b.
  - i. B19...10 (Voltage) set to 5V.
  - j. B9...0 (Maximum Current) set to 500mA.
6. Repeat Step-5 if the Tester does not receive a *GoodCRC* from the UUT in response to *Source\_Capabilities* Message. This requires a delay of *tTypeC\_SendSourceCap* (min to max). This sequence is repeated at least 50 times.
7. The check fails if the UUT does not respond with a *Request* Message. [COMMON.PROC.BU.9#1]
8. The Tester responds with an *Accept* Message to the *Request* Message.
9. When the V<sub>BUS</sub> voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at *tSrcTransReq* max after the reception of the last bit of the *GoodCRC* Message EOP corresponding to the *Accept* Message.
10. An explicit contract is now established.
11. The Tester presents *SinkTxOk* if the test is in PD3 mode. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT. If during this period, the UUT requests a *PR\_Swap*, the Tester sends an *Accept*, and then skips to step 14.
12. The Tester sends a *PR\_Swap* Message.
13. The Tester checks the response. [COMMON.PROC.BU.9#2]

- d. If the UUT responds with a *Wait* Message and the Tester resends *PR\_Swap* every  $tPRSwapWait + tCtsWait$  for  $nCtsWaitCount$  times. Once the Tester has resent *PR\_Swap* for  $nCtsWaitCount$  times, the check fails, and the test ends here.
  - e. The check fails and the test ends here if the UUT responds with a *Reject* Message.
  - f. The check fails and the test ends here if the UUT response with a *Not\_Supported* Message.
14. At  $tSrcTransition$  max (this delay is from the last bit of *GoodCRC* sent in response to the *Accept*), the Tester drives the  $V_{BUS}$  voltage to  $vSafe0V$  within  $tSrcSwapStdby$  max.
15. The Tester sends a *PS\_RDY* at the deadline limit of  $tPSSourceOff$  min, the delay is from the time the last bit of the EOP of the *GoodCRC* corresponding to the *Accept* Message.
16. The tester checks that UUT asserts  $R_p$  before the UUT has applied  $vSafe5V$  [COMMON.PROC.BU.9#3]
17. The Tester checks that the UUT sends *PS\_RDY* only after the UUT has applied  $vSafe5V$ . [COMMON.PROC.BU.9#4]
18. The Tester checks that the *PS\_RDY* from the UUT is within  $tPSSourceOn$  min. [COMMON.PROC.BU.9#5] this delay is from the *GoodCRC* sent by the UUT in response to the *PS\_RDY* sent by the Tester.
19. The Tester checks that the UUT sends a *Source\_Capabilities* Message after  $tSwapSourceStart$  and before  $tFirstSourceCap$  max. [COMMON.PROC.BU.9#6].
20. If there is no PPS APDO in the *Source\_Capabilities*, the test stops here.
21. The Tester responds with a *Request* Message for Programmable Power Supply APDO 5V 1A.
  - a. B26 (Capability Mismatch) = 0b
  - b. B25 (USB Communication Capable) = 0b
  - c. B24 (No USB Suspend) = 1b
  - d. B23 (PD3, Unchunked Extended Messages Supported) = 0b
  - e. B19...9 (Output Voltage) = 5V
  - f. B6...0 (Operating Current) = 1A
22. The check fails if either *Accept* or *Wait* Message is not received, and the test stops here. [COMMON.PROC.BU.9#7]
23. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every  $tSinkRequest$  for  $nCtsWaitCount$  times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after  $nCtsWaitCount$  retries, and the test stops here. [COMMON.PROC.BU.9#10]
24. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within  $tSrcTransReq$  from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [COMMON.PROC.BU.9#8]
25. An explicit contract is now established.
26. The Tester waits  $tCtsAfterLastPoints$  to respond to messages from the UUT.
27. If the UUT is not a Vconn Source, then the Tester sends a *VCONN\_Swap* message and completes the *VCONN\_Swap* using common procedure COMMON.PROC.PD.17. [COMMON.PROC.BU.9#9]

#### COMMON.PROC.BU.10 Bring-up VPD

1. Before starting the test, attach the VPD UUT directly to the tester (a VPD is a Captive Cable device).
2. The test starts in a disconnected state.
3. The Tester applies  $R_p$  to each of the CC1 and CC2 pins at the tester receptacle.
4. The Tester checks that it detects  $R_a$  on one pin (from now on called the  $V_{CONN}$  pin) and  $R_d$  on the other pin (from now on called the CC pin). [COMMON.PROC.BU.10#1]. For any other combination the test ends here.
5. The Tester (Tester is 'Sink with Accessory Support') performs the following:
  - a. keeps  $R_p$  connected to the CC pin.
  - b. waits  $t_{CCDebounce}$ ,
  - c. disconnects  $R_p$  from the  $V_{CONN}$  pin and applies  $V_{CONN}$  (at the voltage specified in the Test Condition) to the  $V_{CONN}$  pin,
6. The Tester waits for  $t_{VCONNStable}$  max from when  $V_{CONN}$  is stable.

#### NOTE:

1. VPD tests need to be run twice, with  $V_{CONN}$  voltage set to 3.0 V, and 5.5V. Additionally, the Eye Diagram test is run at 4.25V.
2. The Tester uses SOP' messages in VPD tests except where otherwise specified.

#### COMMON.PROC.BU.11 Bring-up Source UUT for Captive Cable Test

1. The test starts in a disconnected state.
2. The Tester applies  $R_d$  to each of the CC1 and CC2 pins at the captive cable end.
3. Tester waits for  $R_p$  to appear on exactly one pin. If this does not occur within  $t_{DRP}$ , the check fails. [COMMON.PROC.BU.11#1].
4. The pin on which  $R_p$  appears from now on is called the CC pin. The other pin is referred to as the  $V_{CONN}$  pin.
5. The Tester keeps  $R_d$  connected to the CC pin,
6. Check that  $V_{BUS}$  is detected at  $v_{Safe5V}$  within  $t_{CCDebounce}$  max. [COMMON.PROC.BU.11#2]
7. The Tester disconnects  $R_d$  from the  $V_{CONN}$  pin.
8. Check that the first bit of a *Source\_Capabilities* Message is received from the UUT within  $t_{FirstSourceCap}$  max after  $V_{BUS}$  present ( $v_{Safe5V}$ ). [COMMON.PROC.BU.11#3]
9. The Tester responds with a *Request* Message for 5V at 100mA.
  - a. B30...B28 (Object Position) = 001b
  - b. B27 (Reserved) = 0b
  - c. B26 (Capability Mismatch) = 0b
  - d. B25 (USB Communication Capable) = 0b
  - e. B24 (No USB Suspend) = 1b
  - f. B23 (PD3, Unchunked Extended Messages Supported) = 0b
  - g. B19...10 (Operating Current) = 100mA
  - h. B9...0 (Maximum Operating Current) = 100mA

10. The check fails if the UUT does not respond with either *Accept* or *Wait* Message, and the test stops here. [COMMON.PROC.BU.11#4]
11. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after *nCtsWaitCount* retries, and the test stops here. [COMMON.PROC.BU.11#7]
12. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [COMMON.PROC.BU.11#5]
13. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT.
14. The Tester requests a *VCONN\_Swap*. If this is not accepted the check fails. [COMMON.PROC.BU.11#6]
15. During the *VCONN\_Swap* procedure the value of  $V_{CONN}$  applied is determined by the Test Condition.
16. The Tester sends a BIST request to the UUT using SOP, specifying BIST Test Data.

**NOTE:** The Tester uses SOP' messages in Captive Cable Plug tests except where otherwise specified.

#### COMMON.PROC.BU.12 Bring-up Provider-Consumer as Sink UUT PPS

1. The test starts in a disconnected state.
2. If VIF field *Captive\_Cable* = YES, the Tester applies Rd. Otherwise (VIF field *Captive\_Cable* = No), the Tester applies Rd and Ra.
3. The check fails if the first bit of a *Source\_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after *vSafe5V* present on the  $V_{BUS}$ . [COMMON.PROC.BU.12#1]
4. The Tester responds with a *Request* Message for 5V at 100mA.
  - a. B30...B28 (Object Position) = 001b
  - b. B27 (Reserved) = 0b
  - c. B26 (Capability Mismatch) = 0b
  - d. B25 (USB Communication Capable) = 0b
  - e. B24 (No USB Suspend) = 1b
  - f. B23 (PD3, Unchunked Extended Messages Supported) = 0b
  - g. B19...10 (Operating Current) = 100mA
  - h. B9...0 (Maximum Operating Current) = 100mA



5. The check fails if the UUT does not respond with either *Accept* or *Wait* Message, and the test stops here. [COMMON.PROC.BU.12#2]
6. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after *nCtsWaitCount* retries, and the test stops here. [COMMON.PROC.BU.12#8]
7. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [COMMON.PROC.BU.12#3]
8. An explicit contract is now established.
9. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT. If during this period, the UUT requests a *PR\_Swap*, the Tester sends an *Accept*, and then skips to step 12.
10. The Tester sends a *PR\_Swap* Message.
11. The Tester checks the response. [COMMON.PROC.BU.12#4]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *PR\_Swap* every *tPRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. Once the Tester has resent *PR\_Swap* for *nCtsWaitCount* times, the check fails, and the test ends here.
  - b. If the UUT responds with *Reject* or *Not\_Supported*, the check fails, and the test ends here.
  - c. If the UUT responds with an *Accept* message, proceed to step 12.
12. The Tester draws no more than *iSwapStandby* within *tSrcTransition* min after the last *GoodCRC*.
13. The Tester checks that the UUT asserts *Rd* and sends a *PS\_RDY* Message after the UUT driving *V<sub>BUS</sub>* voltage to *vSafe0V*. [COMMON.PROC.BU.12#5]
14. The Tester checks that the UUT sends *PS\_RDY* within *tPSSourceOff* min after the last bit of *GoodCRC* was sent in response to the received *Accept*. [COMMON.PROC.BU.12#6]
15. The Tester sets *Rp* 3A, then presents *vSafe5V* on *V<sub>BUS</sub>* and then it sends a *PS\_RDY* Message, both within *tNewSRC* max of the last *GoodCRC* sent by the UUT.
16. The Tester transmits *Source\_Capabilities* Message with five PDOs:
  - a. First Fixed Supply PDO
    - i. B31...30 (Fixed Supply) set to 00b.
    - ii. B29 (Dual-Role Power) set to 1b.
    - iii. B28 (USB Suspend Supported) set to 0b.

- iv. B27 (Unconstrained Power) set to 1b.
  - v. B26 (USB Communications Capable) set to 0b.
  - vi. B25 (Dual-Role Data) set to 0b.
  - vii. B24 (PD3, Unchunked Extended Messages Supported) set to 0b.
  - viii. B21...20 (Peak Current) set to 00b.
  - ix. B19...10 (Voltage) set to 5V.
  - x. B9...0 (Maximum Current) set to 3A.
- b. Following 3 Fixed PDOs:
- i. B31...30 (Fixed Supply) set to 00b
  - ii. B29 through B20 set to 0b.
  - iii. B19...10 (Voltage) set to 9V, 15V and 20V.
  - iv. B9...0 (Maximum Current) set to 3A
- c. APDO PPS
- i. B31...30 (Augmented PDO) set to 11b.
  - ii. B29...28 00b – Programmable Power Supply
  - iii. B27 (PPS Power Limited) set to 0b.
  - iv. B24...17 (Maximum Voltage) set to 21V.
  - v. B15...8 (Minimum Voltage) set to 5V.
  - vi. B6...0 (Maximum Current) set to 3A.
17. The check fails if a *Request* Message is not received. [COMMON.PROC.BU.12#7]
18. The Tester responds with an *Accept* Message to the *Request* Message.
19. When the  $V_{BUS}$  voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at *tSrcTransReq* max after the reception of the last bit of the *GoodCRC* Message EOP corresponding to the *Accept* Message.
20. An explicit contract is now established.
21. The Tester presents *SinkTxOK*.
22. If the Sink doesn't request the PPS APDO in 5s from the time Tester presented *SinkTxOK*, the test stops here. If the *Request* Message is received, the Tester responds with an *Accept* Message.



23. When the  $V_{BUS}$  voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at *tSrcTransReq* max after the reception of the last bit of the *GoodCRC* Message EOP corresponding to the *Accept* Message.
24. The Tester presents *SinkTxOK*. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT.
25. If the Tester is not a Vconn Source, then the Tester sends a *VCONN\_Swap* message and completes the *VCONN\_Swap* using common procedure COMMON.PROC.PD.17. [COMMON.PROC.BU.12#8]

### 3.3 Common Procedures

#### 3.3.1 PD2 and PD3 Modes

The common procedures in this subsection are applicable when the test is running in PD2 and PD3 modes.

##### *COMMON.PROC.PD.1 Tester Sends GoodCRC*

**Procedure Applicability:** Unless otherwise specified, the Tester replies with a *GoodCRC* whenever receiving any PD message from the UUT.

1. The Tester sends a *GoodCRC* Message to acknowledge messages received from the UUT. The Tester starts the preamble of *GoodCRC* Message after tInterframeGap min of the last bit of the previously received message.

##### *COMMON.PROC.PD.2 UUT Sent Get\_Source\_Cap*

**Procedure Applicability:** Unless otherwise specified, the Tester runs this procedure whenever receiving *Get\_Source\_Cap* Message from the UUT.

1. The Tester receives a *Get\_Source\_Cap* Message from the UUT. This message is valid except the following conditions: [COMMON.PROC.PD.2#1]
  - a. The check fails if the UUT sends this message before the Tester has established an Explicit Contract
  - b. The check fails if the UUT sends this message within a non-interruptible AMS
  - c. The check fails if the VIF field PD\_Port\_Type = Provider Only
2. There are three scenarios:
  - a. The Tester is a Source
    - i. The Tester responds with *Source\_Capabilities* Message that contains single PDO
      1. B31...30 (Fixed Supply) set to 00b.
      2. B29 (Dual-Role Power) set to 0b if Tester is a Source only; set to 1b if Tester is a Dual-Role Power
      3. B28 (USB Suspend Supported) set to 0b.
      4. B27 (Unconstrained Power) set to 1b.
      5. B26 (USB Communications Capable) set to 0b.
      6. B25 (Dual-Role Data) set to 0b.
      7. B24...22 (Reserved) set to 00b.
      8. B21...20 (Peak Current) set to 00b.
      9. B19...10 (Voltage) set to 5V.

10. B9...0 (Maximum Current) set to 500mA.
  - ii. The check fails if the UUT does not respond with a *Request* Message.  
[COMMON.PROC.PD.2#2]
  - iii. The Tester responds with an *Accept* Message to the *Request* Message.
  - iv. When the  $V_{BUS}$  voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at  $t_{SrcTransReq}$  max after the reception of the last bit *GoodCRC* Message EOP corresponding to the *Accept* Message.
- b. The Tester is a Sink only.
  - i. The Tester responds with a *Reject* Message (in PD2 mode) or *Not\_Supported* (in PD3 mode).
- c. The Tester is a Sink with Dual-Role Power
  - i. The Tester responds with a *Source\_Capabilities* Message for best condition for the *PR\_Swap* to be accepted:
    1. For the first PDO:
      - a. B31...30 (Fixed Supply) set to 00b.
      - b. B29 (Dual-Role Power) set to 1b.
      - c. B28 (USB Suspend Supported) set to 0b.
      - d. B27 (Unconstrained Power) set to 1b.
      - e. B26 (USB Communications Capable) matches VIF field  
USB\_Comms\_Capable
      - f. B25 (Dual-Role Data) matches VIF field  
DR\_Swap\_To\_UFP\_Supported
      - g. B24...22 (Reserved) set to 00b.
      - h. B21...20 (Peak Current) set to 00b.
      - i. B19...10 (Voltage) set to 5V.
      - j. B9...0 (Maximum Current) set to VIF field  
Snk\_PDO\_Max\_Current1
    2. Create the subsequent PDOs using Sink PDO values in the VIF:
      - a. If Snk\_PDO\_Supply\_TypeN = Fixed
        - i. B31...30 (Fixed Supply) set to 00b.
        - ii. B19...10 (Voltage) set to VIF field  
Snk\_PDO\_VoltageN
        - iii. B9...0 (Maximum Current) set to VIF field  
Snk\_PDO\_Op\_CurrentN
      - b. If Snk\_PDO\_Supply\_TypeN = Variable
        - i. B31...30 (Variable Supply) set to 10b.
        - ii. B29...20 (Maximum Voltage) set to VIF field  
Snk\_PDO\_MaxVoltageN
        - iii. B19...10 (Minimum Voltage) set to VIF field  
Snk\_PDO\_MinVoltageN
        - iv. B9...0 (Maximum Current) set to VIF field  
Snk\_PDO\_Op\_CurrentN
      - c. If Snk\_PDO\_Supply\_TypeN = Battery
        - i. B31...30 (Battery Supply) set to 01b.
        - ii. B29...20 (Maximum Voltage) set to VIF field  
Snk\_PDO\_MaxVoltageN

- iii. B19...10 (Minimum Voltage) set to VIF field  
Snk\_PDO\_MinVoltageN
  - iv. B9...0 (Maximum Power) set to VIF field  
Snk\_PDO\_Op\_PowerN
- d. If Snk\_PDO\_Supply\_TypeN = PPS, and in Rev 3
- i. B31...28 (PPS) set to 1100b.
  - ii. B24...17 (Maximum Voltage) set to VIF field  
Snk\_PDO\_MaxVoltageN
  - iii. B15...18 (Minimum Voltage) set to VIF field  
Snk\_PDO\_MinVoltageN
  - iv. B6...0 (Maximum Current) set to VIF field  
Snk\_PDO\_Op\_CurrentN

#### *COMMON.PROC.PD.3 UUT Sent Get\_Sink\_Cap*

Procedure Applicability: Unless otherwise specified, the Tester runs this procedure whenever receiving *Get\_Sink\_Cap* Message from the UUT.

1. The Tester receives *Get\_Sink\_Cap* Message from the UUT. This message is valid except the following conditions: **[COMMON.PROC.PD.3#1]**
  - a. The check fails if the UUT sends this message before the Tester has established an Explicit Contract
  - b. The check fails if the UUT sends this message within a non-interruptible AMS
  - c. The check fails if the VIF field PD\_Port\_Type = Consumer only
2. There are three scenarios:
  - a. The Tester is a Source only.
    - i. The Tester responds with a *Reject* Message
  - b. The Tester is a Source with Dual Role Power
    - i. The Tester responds the following *Sink\_Capabilities* Message with single PDO:
      1. B31...30 (Fixed Supply) set to 00b.
      2. B29 (Dual-Role Power) set to 1b.
      3. B28 (Higher Capability) set to 0b.
      4. B27 (Unconstrained Power) set to 1b.
      5. B26 (USB Communications Capable) set to 0b.
      6. B25 (Dual-Role Data) set to 0b.
      7. B24...20 (Reserved) set to 00000b.
      8. B19...10 (Voltage) set to 5V.
      9. B9...0 (Maximum Current) set to 100mA.
  - c. The Tester is a Sink
    - i. The Tester responds the following *Sink\_Capabilities* Message with single PDO:
      1. B31...30 (Fixed Supply) set to 00b.
      2. B29 (Dual-Role Power) set to 0b if Tester is a Sink only; set to 1b if Tester is a Dual-Role Power
      3. B28 (Higher Capability) set to 0b.

4. B27 (Unconstrained Power) set to 1b.
5. B26 (USB Communications Capable) set to 0b.
6. B25 (Dual-Role Data) set to 0b.
7. B24...23 (Fast Role Swap required USB Type-C Current) set to 00b (Fast Swap not supported)
8. B22...20 (Reserved) set to 000b.
9. B19...10 (Voltage) set to 5V.
10. B9...0 (Maximum Current) set to 100mA.

#### *COMMON.PROC.PD.4 UUT Sent Ping*

Procedure Applicability: Unless otherwise specified, the Tester runs this procedure whenever receiving *Ping* Message from the UUT.

1. The Tester receives *Ping* Message from the UUT and ignores it. This message is valid except the following conditions: [COMMON.PROC.PD.4#1]
  - a. The check fails if the UUT sends this message before the Tester has established an Explicit Contract
  - b. The check fails if the UUT sends this message within a non-interruptible AMS.
  - c. The check fails if the UUT sends this message and the UUT is a Sink

#### *COMMON.PROC.PD.5 UUT Sent PR\_Swap*

Procedure Applicability: Unless otherwise specified, the Tester runs this procedure whenever receiving *PR\_Swap* Message from the UUT.

1. The Tester receives *PR\_Swap* Message from the UUT. This message is valid except the following conditions: [COMMON.PROC.PD.5#1]
  - a. The check fails if the UUT sends this message before the Tester has established an Explicit Contract
  - b. The check fails if the UUT sends this message within a non-interruptible AMS
2. The following are the scenarios (for rejecting a *PR\_Swap*).
  - a. The Tester is a Sink, the Tester responds with a *Reject* Message. The check fails if the VIF field Requests\_PR\_Swap\_As\_Src = No. [COMMON.PROC.PD.5#2]
  - b. The Tester is a Source, the Tester responds with a *Reject* Message. The check fails if the VIF field Requests\_PR\_Swap\_As\_Snk = No. [COMMON.PROC.PD.5#3]

#### *COMMON.PROC.PD.6 UUT Sent VCONN\_Swap*

Procedure Applicability: Unless otherwise specified, the Tester runs this procedure whenever receiving *VCONN\_Swap* message from the UUT.

1. The Tester receives *VCONN\_Swap* message from the UUT. This message is valid except the following conditions: [COMMON.PROC.PD.6#1]
  - a. The check fails if the UUT sends this message before the Tester has established an Explicit Contract
  - b. The check fails if the UUT sends this message within a non-interruptible AMS
2. There are two scenarios:
  - a. The Tester is the initial Vconn Source, and the Tester runs the following procedures:

- i. If the VIF field VCONN\_Swap\_To\_On\_Supported = Yes and the UUT sends a *VCONN\_Swap* message, the Tester responds with an *Accept* Message. Otherwise (the VIF field VCONN\_Swap\_To\_On\_Supported = No and the UUT sends a *VCONN\_Swap* message), the check fails, and the test stops here. [COMMON.PROC.PDU.6#2]
    - ii. The check fails if the UUT does not respond with *PS\_RDY* Message within *tVCONNSourceOn* max from the last bit of the EOP of *GoodCRC* Message corresponding to the *Accept* Message sent. [COMMON.PROC.PD.6#3]
    - iii. The Tester stops sourcing VCONN.
  - b. The UUT is the initial VCONN Source, and the Tester runs the following procedures.
    - i. If the VIF field VCONN\_Swap\_To\_Off\_Supported = Yes and the UUT sends a *VCONN\_Swap* message, the Tester responds with an *Accept* Message. Otherwise (VIF field VCONN\_Swap\_To\_Off\_Supported = No and the UUT sends a *VCONN\_Swap* message), the check fails, and the test stops here. [COMMON.PROC.PD.6#4]
    - ii. The Tester applies VCONN and sends *PS\_RDY* Message.

#### COMMON.PROC.PD.7 UUT Sent Discover Identity Request

**Procedure Applicability:** Unless otherwise specified, the Tester runs this procedure whenever receiving *Discover Identity* Command from the UUT.

1. The Tester receives *Discover Identity* Command from the UUT. This message is valid except the following conditions: [COMMON.PROC.PD.7#1]
  - a. If the test is running in PD2 mode
    - i. If the UUT is a UFP, the check fails if:
      1. The message uses SOP' packet **AND** the message is sent after an explicit contract **OR**
      2. The message uses SOP packet.
    - ii. The check fails if the UUT is a Cable Plug
  - b. If the test is running in PD3 mode
    - i. The check fails if message uses SOP' **AND**
      1. UUT is not supplying VCONN **OR**
      2. it is not VCONN source.
    - ii. the check fails if the UUT is a Cable Plug
  - c. The check fails if the VIF field Attempts\_Discov\_SOP = NO and the UUT has sent using SOP packet.
  - d. The check fails if the UUT sends this message using SOP'' packet.
  - e. The check fails if the UUT sends this message within a non-interruptible AMS.
  - f. The check fails if the UUT sends this message using SOP packet before the Tester has established an Explicit Contract
2. The Tester performs the following checks on the Structured VDM Message Header: [COMMON.PROC.PD.7#2]
  - a. Bits 7...6 (Command Type) is set to 00b.
  - b. Bits 4...0b (Command) is set to 1 (*Discover Identity*)
3. The Tester response depends on the SOP\* Communication:

- a. If a SOP' *Discover Identity* Command is received from the UUT and VIF field Captive\_Cable = NO, the Tester responds with a “Responder ACK” in the following manner(unless otherwise defined in a specific test or section on tests):
    - i. ID Header VDO
      1. B31 (USB Communications Capable as USB Host) set to 0b.
      2. B30 (USB Communications Capable as a USB Device) set to 0b.
      3. B29...27 (SOP' Product Type) set to 011b (Passive Cable)
      4. B26 (Modal Operation Supported) set to 0b.
      5. B25...23 (Reserved) set to 000b.
      6. B22...21 (Connector Type) set to 11b (USB Type-C® Plug)
      7. B20...16 (Reserved) set to 00000b.
      8. B15...0 (USB Vendor ID) set to 0x1A0A.
    - ii. Cert Stat VDO
      - B31..0(XID) set to 0
    - iii. Product VDO
      1. B31..16 (USB Product ID) set to 0x0301.
      2. B15..0 (bcdDevice) set to 0
    - iv. Passive Cable VDO
      1. B31...28 (HW Version) set to 0000b.
      2. B27...24 (FW Version) set to 0000b.
      3. B23...21 (VDO Version) set to 000b (Version 1.0)
      4. B20 (Reserved) set to 0b.
      5. B19...18 (USB Type-C® plug to USB Type-C®/Captive) set to 10b (USB Type-C®)
      6. B17 (EPR Mode Capable) set to 1b.
      7. B16...13 (Cable Latency) set to 0001b (<10ns)
      8. B12...11 (Cable Termination Type) set to 01b (V<sub>CONN</sub> required)
      9. B10...9 (Maximum V<sub>BUS</sub> Voltage) set to 11b (50V)
      10. B8...7 (Reserved) set to 00b.
      11. B6...5 (V<sub>BUS</sub> Current Handling Capability) set to 10b (5A)
      12. B4...3 (Reserved) set to 00b.
      13. B2...0 (USB Highest Speed) set to 100b (USB4 Gen4)
  - b. If a SOP' *Discover Identity* Command is received from the UUT, the Tester responds with a “Responder NAK”.
4. The check fails if a message is received from the UUT before the Tester has sent “Responder ACK” or “Responder NAK”. [\[COMMON.PROC.PD.7#3\]](#)

#### [COMMON.PROC.PD.8 UUT Sent Discover SVIDs Request](#)

**Procedure Applicability:** Unless otherwise specified, the Tester runs this procedure whenever receiving *Discover SVIDs* Command from the UUT.

1. The Tester receives *Discover SVIDs* Command from the UUT. This message is valid except the following conditions: [\[COMMON.PROC.PD.8#1\]](#)
  - a. If the test is running in PD2 mode, the check fails if the UUT is a UFP or a Cable Plug
  - b. If the test is running in PD3 mode, the check fails if the UUT is a Cable Plug
  - c. The check fails if the VIF field Attempts\_Discov\_SOP = NO



- d. The check fails if the UUT sends this message using SOP'' packet.
  - e. The check fails if the UUT sends this message within a non-interruptible AMS.
  - f. The check fails if the UUT sends this message before the Tester has established an Explicit Contract
2. The Tester performs the following checks on the Structured VDM Message Header: [COMMON.PROC.PD.8#2]
  - a. Bits 7...6 (Command Type) is set to 00b.
  - b. Bits 4...0 (Command) is set to 2 (Discover SVIDs)
3. The Tester responds with a "Responder NAK". The check fails if a message is received from the UUT before the Tester has sent "Responder NAK". [COMMON.PROC.PD.8#3]

#### COMMON.PROC.PD.9 UUT Sent Attention

Procedure Applicability: Unless otherwise specified, the Tester runs this procedure whenever receiving *Attention* Command from the UUT.

1. The Tester receives *Attention* Command from the UUT. This message is valid except the following conditions: [COMMON.PROC.PD.9#1]
  - a. In PD2, the check fails if the UUT is a DFP or a Cable Plug
  - b. In PD3, the check fails if the UUT is a Cable Plug
  - c. The check fails if VIF field Modal\_Operation\_Supported\_SOP = NO
  - d. The check fails if the UUT sends this message using SOP' or SOP'' packet.
  - e. The check fails if the UUT sends this message within a non-interruptible AMS.
  - f. The check fails if the UUT sends this message before the Tester has established an Explicit Contract
2. The Tester performs the following checks on the Structured VDM Message Header: [COMMON.PROC.PD.9#2]
  - a. Bits 7...6 (Command Type) is set to 00b.
  - b. Bits 4...0b (Command) is set to 6 (*Attention*)
3. The Tester ignores the *Attention* Command and does not send a response.

#### COMMON.PROC.PD.10 UUT Sent Request

Procedure Applicability: Unless otherwise specified, the Tester runs this procedure whenever receiving *Request* Message from the UUT.

1. The Tester receives *Request* Message from the UUT. This message is valid except the following conditions: [COMMON.PROC.PD.10#1]
  - a. The check fails if an Implicit Contract is in place and the message was not in response to a *Source\_Capabilities* Message from the Tester.
  - b. The check fails if the UUT sends this message within a non-interruptible AMS.
  - c. The check fails if the UUT is a Source
  - d. The check fails if the UUT is a Sink and in the EPR mode.
2. The Tester responds with an *Accept* Message to the *Request* Message.
3. When the V<sub>BUS</sub> voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at *tSrcTransReq* max after the reception of the last bit *GoodCRC* Message EOP corresponding to the *Accept* Message.

#### COMMON.PROC.PD.11 UUT Sent Source\_Capabilities

**Procedure Applicability:** Unless otherwise specified, the Tester runs this procedure whenever receiving *Source\_Capabilities* Message from the UUT.

1. The Tester receives *Source\_Capabilities* Message from the UUT. This message is valid except the following conditions: [COMMON.PROC.PD.11#1]
  - a. The check fails if the UUT is a Sink, and the Tester has not previously sent *Get\_Source\_Cap*.
  - b. The check fails if PD\_Port\_Type is set to Consumer Only
  - c. The check fails if the UUT sends this message within a non-interruptible AMS.
2. The procedure ends here if the UUT is a Sink. The Tester sends a *Request* Message for 5V at 100mA.
  - a. B30 (Object Position) = 001b
  - b. B27 (Reserved) = 0b
  - c. B26 (Capability Mismatch) = 1b
  - d. B25 (USB Communication Capable) = 0b
  - e. B24 (No USB Suspend) = 1b
  - f. B19...10 (Operating Current) = 100mA
  - g. B9...0 (Maximum Operating Current) = 100mA
3. The check fails if either the *Accept* or *Wait* Message is not received, and the test stops here [COMMON.PROC.PD.11#2]
4. If the UUT responds with the *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until the *Accept* Message is received. The check fails if the UUT responds with the *Wait* Message after *nCtsWaitCount* retries, and the test stops here. [COMMON.PROC.PD.11#4]
5. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted the *Accept* Message. [COMMON.PROC.PD.11#3]

#### COMMON.PROC.PD.12 UUT Sent the DR\_Swap Message

**Procedure Applicability:** Unless otherwise specified, the Tester runs this procedure whenever receiving *DR\_Swap* Message from the UUT.

1. The Tester receives the *DR\_Swap* Message from the UUT. This message is valid except the following conditions: [COMMON.PROC.PD.12#1]
  - a. The check fails if the UUT sends this message before the Tester has established an Explicit Contract
  - b. The check fails if the UUT sends this message within a non-interruptible AMS.
  - c. The check fails if the UUT is a UFP and the VIF field  
DR\_Swap\_To\_DFP\_Supported = No
  - d. The check fails if the UUT is a DFP and the VIF field  
DR\_Swap\_To\_UFP\_Supported = No

COMMON.PROC.PD.13-16 have been renamed and re-allocated to other sections.

#### COMMON.PROC.PD.17 Tester Sent VCONN\_Swap Message



Procedure Applicability: The Tester runs this procedure wherever COMMON.PROC.PD.17 is referred to in the test procedure.

1. There are two scenarios:

a. The Tester is the initial Vconn Source, and the Tester runs the following procedures:

i. The Tester checks the response. [COMMON.PROC.PD.17#1]

1. If the UUT responds with a *Wait* Message, the Tester checks that VIF parameters Type\_C\_Source\_VCONN and VCONN\_Swap\_To\_ON\_Supported are set to Yes and resends *VCONN\_Swap* after *tVCONNSourceWait* + *tCtsWait* if the Tester hasn't retried for *nCtsWaitCount* times. If one of the specified VIF parameters is set to No or the number of retries to send *VCONN\_Swap* exceeds *nCtsWaitCount* times, the check fails, and the test stops here.
2. If the UUT responds with a *Reject* Message, the Tester checks that the UUT is either
  - a. Provider or DRP,
  - b. Consumer in the PD2 mode, or
  - c. Consumer and VConn\_Swap\_To\_On\_Supported is set to Yes and in the PD3 mode.the common procedure ends here, and the test continues unless otherwise specified in the test. Else, the check fails, and the test stops here.
3. If the UUT responds with a *Not\_Supported* Message, the Tester checks that the UUT is Consumer and VIF parameter VCONN\_Swap\_To\_On\_Supported is set to No, the common procedure ends here, and the test continues. If the UUT is not Consumer or VCONN\_Swap\_To\_On\_Supported is set to Yes, the check fails, and the test stops here.
4. If the UUT responds with the *Accept* Message, the Tester checks the VIF parameters Type\_C\_Source\_VCONN and VCONN\_Swap\_To\_On\_Supported are set to Yes. If VCONN\_Swap\_To\_On\_Supported is set to No or both parameters are set to No, the check fails, and the test stops here.
5. If the UUT responds with any other message, the check fails, and the test ends here.

ii. The check fails if the UUT does not respond with *PS\_RDY* Message within *tVCONNSourceOn* max from the last bit of the EOP of the received *Accept* Message. [COMMON.PROC.PD.17#2]

iii. The Tester stops sourcing VCONN.

b. The UUT is the initial VCONN Source, and the Tester runs the following procedures.

- i. The Tester checks the response. [COMMON.PROC.PD.17#3]
  1. If the UUT responds with the *Wait* Message, the Tester checks that either
    - a. both VIF parameters Type\_C\_Source\_VCONN and VCONN\_Swap\_To\_Off\_Supported are set to Yes, or
    - b. VIF parameter Type\_C\_Source\_VCONN is set to No, VCONN\_Swap\_To\_Off\_Supported is set to Yes and the product is either Provider or DRP. If both conditions are not met, the check fails, and the test stops here.

The Tester resends *VCONN\_Swap* after *tVCONN\_SwapWait* + *tCtsWait* if the Tester hasn't retried for *nCtsWaitCount* times. If one of the specified VIF parameters is set to No or the number of retries to send *VCONN\_Swap* exceeds *nCtsWaitCount* times, the check fails, and the test stops here.
  2. If the UUT responds with a *Reject* Message
    - a. In PD3 mode, the check fails, and the test ends here.
    - b. In PD2 mode, the Tester checks that both VIF parameters Type\_C\_Source\_VCONN and VCONN\_Swap\_To\_On\_Supported are set to No and the product is either Provider or DRP. In this case the common procedure ends here, and the test continues unless otherwise specified in the test. Else, if either of VIF parameters is set to Yes or both, or the product is Consumer, the check fails, and the test ends here.
  3. If the UUT responds with a *Not\_Supported* Message
    - a. In PD 2 mode, the check fails, and the test ends here.
    - b. In PD3 mode, the Tester checks that both VIF parameters Type\_C\_Source\_VCONN and VCONN\_Swap\_To\_On\_Supported are set to No and the product is either Provider or DRP. In this case the common procedure ends here, and the test continues unless otherwise specified in the test. Else, if either of VIF parameters is set to Yes or both, or the product is Consumer, the check fails, and the test ends here.
  4. If the UUT responds with the *Accept* Message, the Tester checks that either
    - a. both VIF parameters Type\_C\_Source\_VCONN and VCONN\_Swap\_To\_Off\_Supported are set to Yes, or
    - b. VIF parameter Type\_C\_Source\_VCONN is set to No, VCONN\_Swap\_To\_Off\_Supported is set to Yes and the product either Provider or DRP.

If any of the conditions is not met, the check fails, and the test stops here.

5. If the UUT responds with any other message, the check fails, and the test ends here.
- ii. The Tester sends a *PS\_RDY* Message. The Tester checks that the UUT does not present VCONN at the non-CC line within *tVCONNSourceOff*.  
[COMMON.PROC.PD.17#4]

#### COMMON.PROC.PD.18 Tester sent Request with Capability Mismatch for Full Capabilities Testing

**Procedure Applicability:** The Tester runs this procedure wherever COMMON.PROC.PD.18 is referred to in the test procedure.

1. The tester sends a *Request* Message as defined below:
  - a. B30...B28 (Object Position) = Last Source Fixed PDO
  - b. B27 (Reserved) = 0b
  - c. B26 (Capability Mismatch) = 1b
  - d. B25 (USB Communication Capable) = 0b
  - e. B24 (No USB Suspend) = 1b
  - f. B19...10 (Operating Current) = Maximum Current in Source PDO
  - g. B9...0 (Maximum Operating Current) = 5A.
2. If the *Wait* Message is received, the Tester re-sends the *Request* Message every *tSinkRequest* up to a maximum of *nBusyCount* times or until *Accept* is received.
3. If an *Accept* Message is not received, the check fails, and the test stops here.
4. If the first Preamble bit of *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [COMMON.PROC.PD.18#1]
5. The Tester responds to any *Get\_Sink\_Cap* Message with the *Sink\_Capabilities* Message using the values found in the UUT VIF Source PDO values. It also responds to any *Get\_Sink\_Cap\_Extended* Message using the values defined in **Table 16** Sink Capabilities Extended Message.
6. If *Source Capabilities* are received with less than the *Sink Capabilities* (same as VIF Source Capabilities) and *tCapabilitiesMismatchResponse* has not elapsed, then the Tester goes back and repeats the *Request* Message in step 1.a with the Capability Mismatch bit =1.
7. Within *tCapabilitiesMismatchResponse*,
  - a. If the *Source Capabilities* matching those in the VIF have been received by the Tester, it sends the *Request* Message defined in Step 1 with Capability Mismatch bit = 0. The check passes and proceeds to the next step.  
[COMMON.PROC.PD.18#2]

Check 1{

- b. If the *Source Capabilities* not matching in the VIF have been received by the Tester. [COMMON.PROC.PD.18#4]
  - i. In the PD2 mode, proceed to the next step.

- ii. In the PD3 mode, the Tester sends the *Get\_Source\_Info* Message to the UUT and checks the response.
  - 1. If the PDP of the received *Source\_Capabilities* is greater or equal to SIDO2 B8..0 (Port Guaranteed PDP), the test passes and proceeds to the next step.
  - 2. If the PDP of the received *Source\_Capabilities* is greater or equal to the minimum of either Port Present PDP in the *Source\_Info* Message or Sink Maximum PDP field in the *Sink\_Capabilities\_Extended* Message, the test passes and proceeds to the next step, else the test fails and stops here.

Check 2 {

- c. If no Source Capabilities have been received by the Tester  
[COMMON.PROC.PD.18#5]

- i. In the PD2 mode, proceed to the next step.
- ii. In the PD3 mode,
  - 1. The Tester sends the *Get\_Source\_Info* Message to the UUT and checks the response.
  - 2. If SIDO1 B15...8 (Port Present PDP) equals to B7...0 (Port Reported PDP), the check passes and test proceeds to the next step.
  - 3. If any of the following conditions are met, the check fails, and the test stops here.
    - a. SIDO1 B15...8(Port Present PDP) is not equal to B7...0 (Port Reported PDP), **or**
    - b. the UUT does not reply with *Source\_Info*, **or**
    - c. the UUT responds with a *Not Supported* Message.

} // Check 2

**NOTE:** It is expected that the internal Source state of the UUT is constant during this check, and it does not cause a change in Port Present PDP or Reported PDP .

- 8. If the *Wait* Message is received, the Tester re-sends the *Request* Message every *tSinkRequest* up to a maximum of *nBusyCount* times or until *Accept* is received.
- 9. If *Accept* Message is not received, the check fails, and the test stops here.
- 10. If the first Preamble bit of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [COMMON.PROC.PD.18#3]

### 3.3.2 PD3 Mode

The common procedures in this subsection are applicable only when the test is running in PD3.

#### COMMON.PROC.PD3.1 Sink Start an AMS

**Procedure Applicability:** Unless otherwise specified, the Tester runs this procedure whenever it needs to start an AMS.

- 1. The Tester as a Sink monitors the Rp value to start an AMS:
  - If the UUT presents *SinkTxOK*, the Sink proceeds to send message (and start an AMS) as described in the test.

### COMMON.PROC.PD3.2 UUT Sent *EPR\_Source\_Capabilities* Message

**Procedure Applicability:** Unless otherwise specified, the Tester runs this procedure whenever receiving an *EPR\_Source\_Capabilities* Message from the UUT.

1. The Tester receives the *EPR\_Source\_Capabilities* Message from the UUT. This message is valid except for the following conditions: [COMMON.PROC.PD3.2#1]
  - a. When UUT is in SPR mode, the check fails if the Tester has not previously sent *EPR\_Get\_Source\_Cap*.
  - b. The check fails if PD\_Port\_Type is set to Consumer Only
  - c. The check fails if the UUT sends this message within a non-interruptible AMS.
  - d. The check fails if the VIF field EPR\_Supported\_As\_Src is set to No and the test stops here.
2. The procedure ends here if the UUT is a Sink or in the SPR mode. Otherwise, the Tester sends an *EPR\_Request* Message.
  - a. Object Position
    - i. For *EPR\_Source\_Capabilities* Message with no EPR PDO, Object Position = 001b
    - ii. For *EPR\_Source\_Capabilities* Message with EPR PDOs, Object Position = 1000b
  - b. B27 (Reserved) = 0b
  - c. B26 (Capability Mismatch) = 1b
  - d. B25 (USB Communication Capable) = 0b
  - e. B24 (No USB Suspend) = 1b
  - f. B19...10 (Operating Current) = 100mA
  - g. B9...0 (Maximum Operating Current) = 100mA
3. The check fails if an *Accept* Message is not received. [COMMON.PROC.PD3.2#2]
4. The check fails if the first Preamble bit of *PS\_RDY* Message is not received within EPR's mode *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [COMMON.PROC.PD3.2#3]

### COMMON.PROC.PD3.3 UUT Sent *EPR\_Get\_Source\_Cap*

Procedure Applicability: Unless otherwise specified, the Tester runs this procedure whenever receiving an *EPR\_Get\_Source\_Cap* Message from the UUT.

1. The Tester receives an *EPR\_Get\_Source\_Cap* Message from the UUT. This message is valid except for the following conditions: [COMMON.PROC.PD3.3#1]

- a. The check fails if the UUT sends this message before the Tester has established an Explicit Contract
- b. The check fails if the UUT sends this message within a non-interruptible AMS.
- c. The check fails if the VIF field PD\_Port\_Type = Provider Only
- d. The check fails if the VIF field *EPR\_Supported\_As\_Snk* is set to No and the test stops here.

2. There are three scenarios:

- a. The Tester is a Source only or Sink with Dual role power, and if the “EPR Mode Capable” bit is set to 0 in the *Source\_Capabilities* Message, Tester sends *Not\_Supported* Message
- b. The Tester is a Source only or Sink with Dual role power, and if the “EPR Mode Capable” bit is set to 1 in the *EPR\_Source\_Capabilities* Message that contains seven PDO’s and four EPR\_PDOs (240W PDP)
  1. B31...30 (Fixed Supply) set to 00b.
  2. B29 (Dual-Role Power) set to 0b if Tester is a Source only; set to 1b if Tester is a Dual-Role Power
  3. B28 (USB Suspend Supported) set to 0b.
  4. B27 (Unconstrained Power) set to 1b.
  5. B26 (USB Communications Capable) set to 0b.
  6. B25 (Dual-Role Data) set to 0b.
  7. B24 (Unchunked Extended messages Supported) set to 0b or 1b depending on test condition.
  8. 23 set to 1b (*EPR\_Mode* Capable)
  9. 22 (Reserved) set to 0b

10. B21...20 (Peak Current) set to 00b.

PDO1: 9. B19...10 (Voltage) set to 5V.

10. B9...0 (Maximum Current) set to 3000mA

PDO2: 9. B19...10 (Voltage) set to 9V.

10. B9...0 (Maximum Current) set to 3000mA

PDO3: 9. B19...10 (Voltage) set to 15V.

10. B9...0 (Maximum Current) set to 3000mA

PDO4: 9. B19...10 (Voltage) set to 20V.

10. B9...0 (Maximum Current) set to 5000mA

PDO5: B31..30 (APDO) set to 11b

B29..28 (SPR AVS) set to 10b

B27..26 (Peak Current) set to 00b

B25..20 (Reserved) set to 00b

B19..10 (9-15V current) set to 3000mA

B9..0 (15-20v current) set to 5000mA

PDO6: All fields set to 0.

PDO7: All fields set to 0.

PDO8: 9. B19...10 (Voltage) set to 28V.

10. B9...0 (Maximum Current) set to 5000mA

PDO9: 9. B19...10 (Voltage) set to 36V.

10. B9...0 (Maximum Current) set to 5000mA

PDO10: 9. B19...10 (Voltage) set to 48V.

10. B9...0 (Maximum Current) set to 5000mA

PDO11: 1. B31...B30 (Augmented) set to 11b.

2. B29..B28 (EPR Adjustable Voltage Supply) 01b

3. B27..B26 (Peak Current) set to 00b
4. B25..B17(maximum voltage) set to 48V
5. B16(reserved) set to 0b
6. B15..B8 (minimum voltage) set to 15V
7. B7..B0 (PDP) set to 240

Tester checks below steps only in EPR mode:

- i. The check fails if the UUT does not respond with an *EPR\_Request* Message. [COMMON.PROC.PD3.3#2]
  - ii. The check fails if requested voltage is not part of VIF PDO list or PDP requested is greater than VIF field PD\_power\_as\_Snk [COMMON.PROC.PD3.3#3]
  - iii. The Tester responds with an *Accept* Message to the *EPR\_Request* Message.
  - iv. When the  $V_{BUS}$  voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at EPR's mode *tSrcTransReq* max after the reception of the last bit of the *GoodCRC* Message EOP corresponding to the *Accept* Message.
- c. The Tester is a Sink only.
- i. The Tester responds with a *Reject* Message.

#### COMMON.PROC.PD3.4 UUT Sent *EPR\_Request*

Procedure Applicability: Unless otherwise specified, the Tester runs this procedure whenever receiving an *EPR\_Request* Message from the UUT.

1. The Tester receives an *EPR\_Request* Message from the UUT. This message is valid except for the following conditions: [COMMON.PROC.PD3.4#1]
  - a. The check fails if an Implicit Contract is in place.
  - b. The check fails if the UUT sends this message within a non-interruptible AMS.
  - c. The check fails if the UUT is a Source or if the UUT is in SPR mode.
  - d. The check fails if the VIF field *EPR\_Supported\_As\_Snk* is set to No and the test stops here.



2. If the Tester is in EPR mode, the Tester responds with an *Accept* Message to the *EPR\_Request* Message, otherwise the Tester sends *Not\_Supported* Message, and the test stops here.
3. When the  $V_{BUS}$  voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at EPR's mode *tSrcTransReq* max after the reception of the last bit of the *GoodCRC* Message EOP corresponding to the *Accept* Message.

#### *COMMON.PROC.PD3.5 Tester Sent EPR\_Mode (Enter)*

Procedure Applicability: Unless otherwise specified, the Tester runs this procedure whenever the Tester is a Sink and tries to enter the EPR mode (0x01).

1. The Tester checks the UUT response as following:
  - a. If the UUT responds with the Action field of the *EPR\_Mode* Message set to the value 4(*Enter failed*), the Tester performs additional checks:
    - i. If the Data field of the *EPR\_Mode* Message set to value 5(*EPR Mode Capable bit not set in PDO*), the Tester checks VIF *EPR\_Supported\_As\_Src*. If *EPR\_Supported\_As\_Src* = Yes, the check fails, and the test stops here. Else, the check passes, and the test stops here. [COMMON.PROC.PD3.5#1]
    - ii. If the Data field of the *EPR\_Mode* Message set to value 4(*Source unable to Enter EPR Mode at this time*) (If it is the second instance of the response, the check fails, and the test stops here).
      1. If VIF Has *\_Invariant\_PDOs* set to Y, the Tester fails the check, and the test stops here [COMMON.PROC.PD3.5#2]
      2. Else, the Tester checks if the UUT can provide a higher power:
        - a. The Tester sends the *Request* Message with the Capability Mismatch bit (B26) set to 1b, and Maximum Operating Power field value set to the VIF *Src\_PDO\_Max\_Power* Parameter
        - b. The check fails if either *Accept* or *Wait* Message is not received, and the test stops here. [COMMON.PROC.PD3.5#3]
        - c. If the UUT responds with the *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until the *Accept* Message is received. The check fails if the UUT responds with the *Wait* Message after *nCtsWaitCount* retries, and the test stops here. [COMMON.PROC.PD3.5#4]
        - d. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted the *Accept* Message. [COMMON.PROC.PD3.5#5]

- f. If Source Capabilities are received with less than the VIF specified Source Capabilities PDOs and 2 seconds have not elapsed after *PS\_RDY*, then the Tester goes back to step 1.a.ii.2.a. (specify which *PS\_RDY* – first explicit contract) [COMMON.PROC.PD3.5#6]
    - g. Within 2 seconds the Source Capabilities matching those in the VIF should have been received by the Tester, else the check fails, and the test stops here. [COMMON.PROC.PD3.5#7]
    - e. Otherwise, the Tester sends *EPR\_Mode*(Enter), and goes back to step 1.
      - iii. Else, the check fails, and the test stops here. [COMMON.PROC.PD3.5#8]
  - b. If the UUT responds with the *Not\_Supported* Message, the Tester checks the VIF field *EPR\_Supported\_As\_Source* is set to No, the check passes, and the test stops here. [COMMON.PROC.PD3.5#9]
  - c. Else, the Tester checks that the UUT responds with the Action field of the *EPR\_Mode* Message is set to the 2(*Enter Acknowledge*) value and the Data field is set to 0x0.
    - i. If the Message is not received within *tEnterEPR* measured from the last bit of the *GoodCRC* in response to the *EPR\_Mode* (Enter) Message, the test fails, and the test stops here. [COMMON.PROC.PD3.5#10]
    - ii. Else, continue to the next step.
- 2. The Tester performs below checks: [COMMON.PROC.PD3.5#11]
  - a. If the UUT sends the *VCONN\_Swap* Message, the Tester:
    - iii. checks that the VIF field *Captive\_Cable* is NO and the UUT is not a *VConn\_Source*. If the check fails, the test stops here.
    - iv. completes *VConn\_Swap* sequence using COMMON.PROC.PD.6,
    - v. continue to step 2.b. to check the next message.
  - b. If the UUT sends SOP' *Discover Identity* Command, the Tester checks that *VConn* present at non-CC line and replies as a passive cable (See Sec 7.1.1 – Test procedure for *EPR Source Capable UUT*). Continue to step 2.c.
  - c. If the UUT sends the *EPR\_Mode* Message with the Action field set to 3(*Enter\_Succeeded*) within *tEnterEPR* (measured from the last bit of the *GoodCRC* in response to *EPR\_Mode* with the Action field set to 2(*Enter Acknowledge*)) without sending the SOP' *Discover Identity* Command, the Tester checks that the UUT discovers the cable capability prior to the explicit contract or within *tEPRSourceCableDiscovery* measured from the last bit of *GoodCRC* of the *PS\_RDY* Message as a part of the explicit contract. Continue to step 3.
  - d. Else, the check fails, and the test stops here.
- 3. The Tester checks that the UUT sends *EPR\_Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* (Enter\_Succeeded) Message. [COMMON.PROC.PD3.5#12]
- 4. The Tester checks that the UUT establishes *EPR* contract using common procedure COMMON.PROC.PD3.2 [COMMON.PROC.PD3.5#13]

### COMMON.PROC.PD3.6 UUT Sent *EPR\_Mode* (Enter)

Procedure Applicability: The Tester runs this procedure wherever COMMON.PROC.PD3.6 is referred in the test procedure.

1. The Tester receives an *EPR\_Mode* Message with the Action field set to 1(Enter) and Data field set to UUT's operational PDP from the UUT. This message is valid except the following conditions: [COMMON.PROC.PD3.6#1]
  - a. The check fails if the UUT sends this message before the Tester has established an Explicit Contract
  - b. The check fails if the UUT sends this message within a non-interruptible AMS.
  - c. The check fails if the UUT is a Source.
  - d. The check fails if the VIF field *EPR\_Supported\_As\_Snk* is set to No.
2. There are three scenarios: [COMMON.PROC.PD3.6#2]
  - a. The Check fails if the *EPR Mode Capable* bit was not set in the most recent RDO, then the Tester sends an *EPR\_Mode* Message with the Action field set to 4(*Enter Failed*) and the Data field set to 3(*EPR Mode Capable bit not set in RDO*) and the test stops here.
  - b. The Check fails if the *EPR Mode Capable* bit was not set in the most recent 5V fixed PDO from the Tester, then the Tester sends an *EPR\_Mode* Message with the Action field set to 4(*Enter Failed*) and the Data field set to 5(*EPR Mode Capable bit not set in PDO*) and the test stops here.
  - c. The Tester sends an *EPR\_Mode* Message with the Action field set to 2(*Enter Acknowledged*) and the Data field set to 0.
3. If the tester is not the *V<sub>CONN</sub>* source, then the Tester sends a *VCONN\_Swap* message and completes the *VCONN\_Swap* using common procedure COMMON.PROC.PD.17. [COMMON.PROC.PD3.6#3]
4. The check fails if the tester fails to become the *V<sub>CONN</sub>* source, then the tester sends an *EPR\_Mode* Message with the Action field set to 4(*Enter failed*) and the Data field set to 2(*Source failed to become VCONN source*) and the test stops here. [COMMON.PROC.PD3.6#4]
5. If VIF field *Captive\_Cable* is Yes, the Tester sends SOP' *Discover Identity* Command and checks the below fields in the response message, [COMMON.PROC.PD3.6#5]
  - a. *V<sub>BUS</sub>* Current Handling Capability set to 10b(5A)
  - b. Maximum *V<sub>BUS</sub>* Voltage set to 11b(50V)
  - c. *EPR Mode Capable* field set to 1b (*EPR Mode Capable*)
  - d. The check fails if the UUT fails to respond to the *Discover Identity* Command or is not *EPR* capable, then the Tester sends an *EPR\_Mode* Message with the Action field set to 4(*Enter failed*) and the Data field set to 1(*Cable not EPR capable*) and the test stops here.
6. The Tester sends an *EPR\_Mode* Message with Action field set to 3(*Enter\_Succeeded*) and Data field set to 0 within *tEnterEPR* from the *EPR\_Mode* (Enter) Message.
7. The Tester sends *EPR\_Source\_Capabilities* Message (as per the COMMON.PROC.PD3.3 step 2.b.i) within 50ms from the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* (Enter\_Succeeded) Message and the Tester checks that the UUT responds with *EPR\_Request* Message. [COMMON.PROC.PD3.6#6]

8. The Tester checks that the UUT establishes the EPR contract negotiation using COMMON.PROC.PD3.4. [COMMON.PROC.PD3.6#7]

#### COMMON.PROC.PD3.7 Tester Sends EPR\_KeepAlive Message

**Procedure Applicability:** This procedure needs to run every time when the Tester is a Sink and operates in the EPR mode.

1. Tester sends *EPR\_KeepAlive* Message for every  $t_{SinkEPRKeepAlive}$  max it has stayed at PE\_SNK\_Ready State.
2. Tester checks that the UUT responds with the *EPR\_KeepAlive\_Ack* Message [COMMON.PROC.PD3.7#1]

## 4 Physical Layer Specific Tests

This chapter describes tests that primary cover Chapters 6 and 8 requirements in the USB PD specifications.

Table 9 Receiver Test Signaling Table

	TX_NORMAL	TX_GROUP_1	TX_GROUP_2	TX_GROUP_3
High Level (nom)	1100 mV	1200 mV	790 mV	1290 mV
Low Level (nom)	25mV	0 mV	-250mV	250mV
Bit Rate	300 kb/s	270 kb/s	330 kb/s	330 kb/s
Rise/Fall Time	735 ns	735 ns	735 ns	735 ns

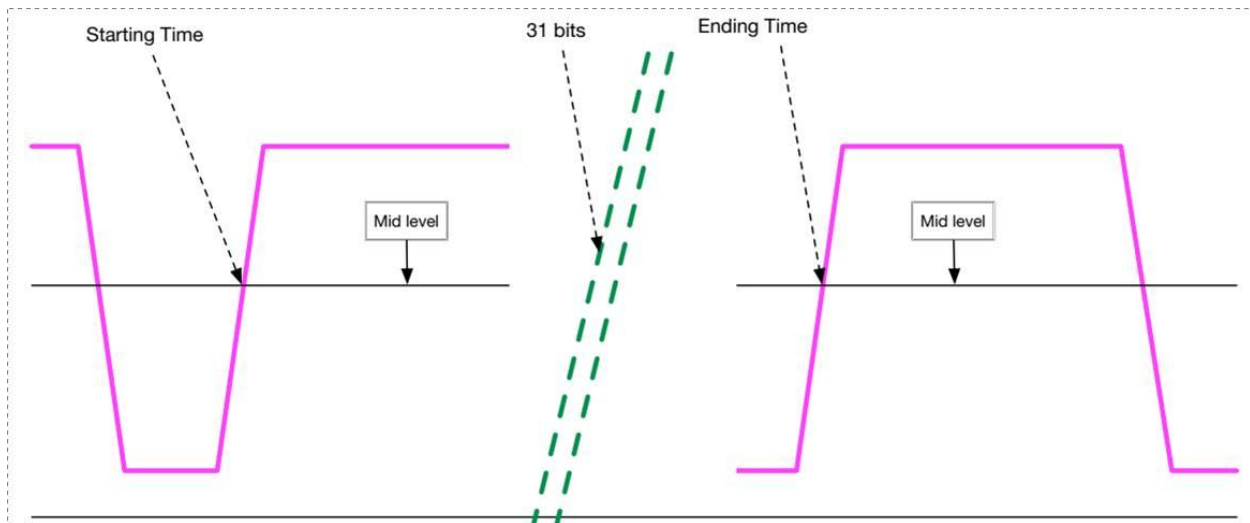


Figure 1: using mid-level to find the reference unit interval

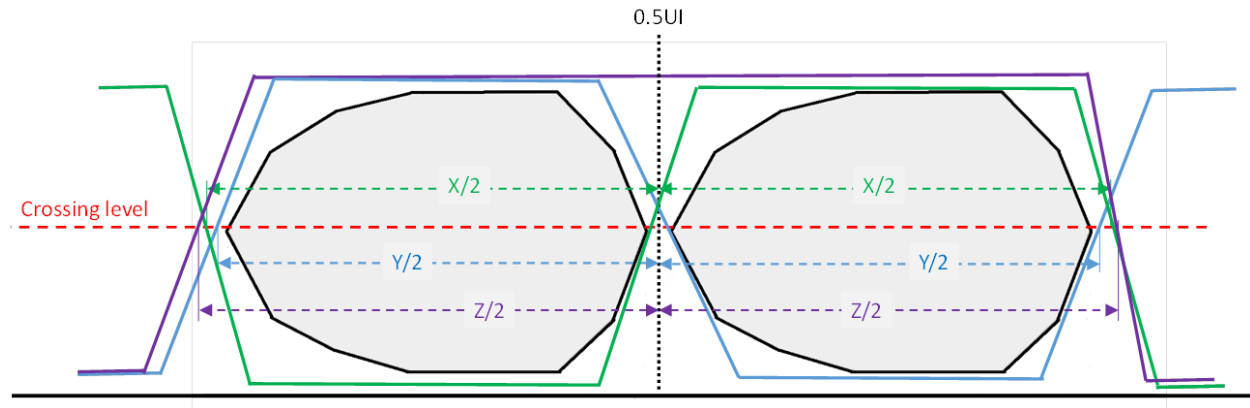


Figure 2: Example of producing eye diagram

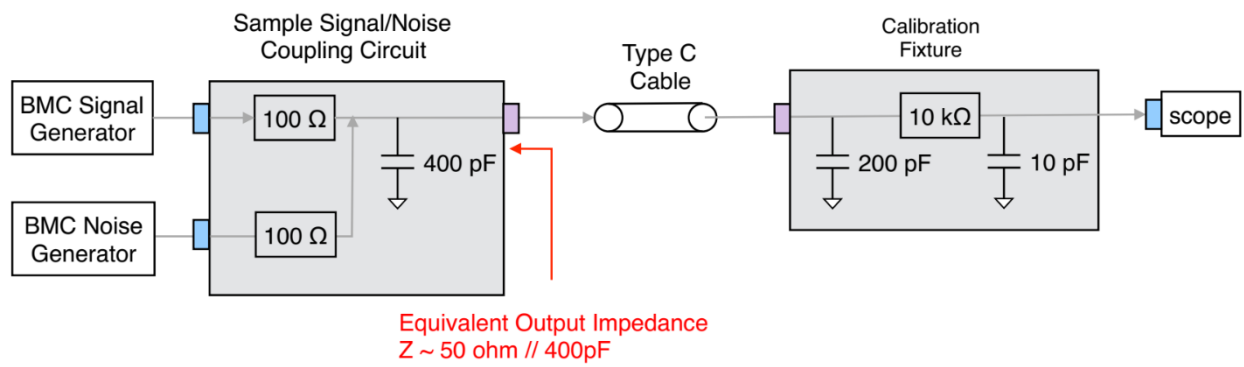


Figure 3: Noise injection generator and calibration fixture setup

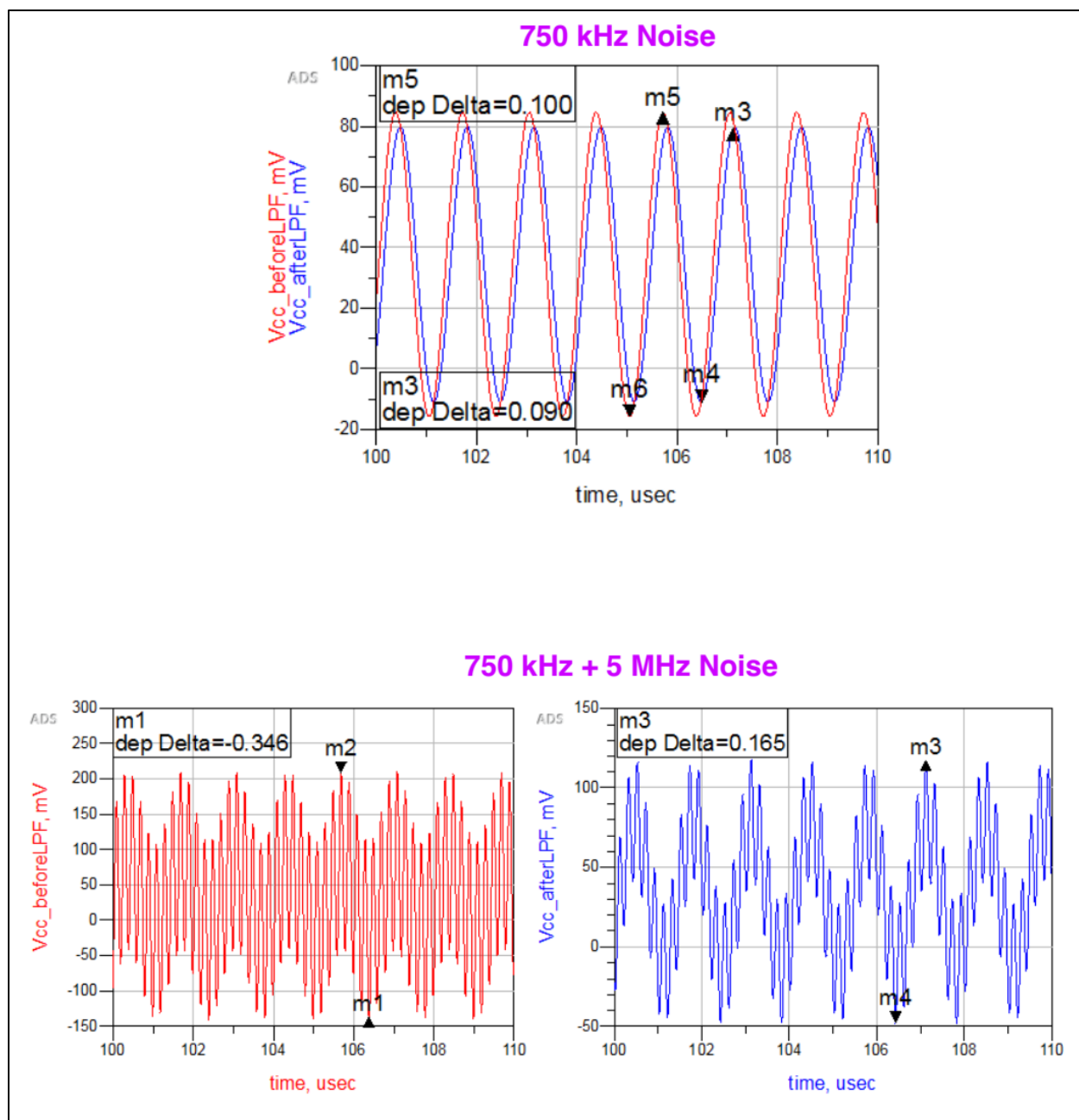


Figure 4: Two-tone noise calibration

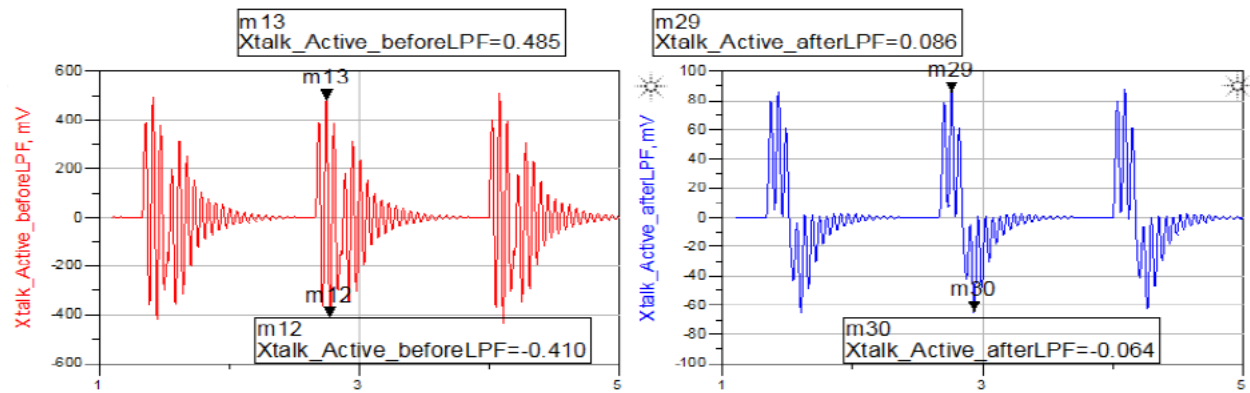


Figure 5: AWG measurement waveforms

## 4.1 Test Procedures for All UUTs

### 4.1.1 PD3 Mode

The tests in this subsection are run only in PD3 mode.

The tests in this subsection are applicable to all UUTs.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (ie, a Provider/Consumer or Consumer/Provider)
- The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### *TEST.PD.PHY.ALL.1 Transmit Bit Rate and the Drift*

**Description:** The Tester checks that the UUT transmitted data meets the bit rate requirements, and it correctly implements BIST Carrier Mode 2.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, P/C	DRP, C/P	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3ChkdSrc</i>		✓			
<i>Rev3ChkdSnk</i>	✓		✓		
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

**Test Procedures:**

1. There are 6 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider or DRP. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Provider Only or Provider/Consumer. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - c. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - d. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10



- e. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
- f. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a BIST request to the UUT, specifying “BIST Carrier Mode”.
3. The Tester creates a histogram with at least 100 bins of the total voltage swing for the bit stream collected from the BIST request. For example, 100 bins, the voltage absolute max is 1.1V and absolute min is 0V; then the top bin is count of voltages between 1.089V and 1.1V; bottom bin is count of voltages between 0V and 0.011V
4. The Tester searches for the bin with highest count between Y5TX (0.5625V) and the maximum; and the middle voltage of the bin is “high level”.
5. The Tester searches for the bin with highest count between Y5TX (0.5625V) and the minimum; and the middle voltage of the bin is “low level”.
6. The average between “high level” and “low level” is the mid-level.
7. The Tester uses the mid-level to find *fBitRateRef* (refer to [Figure 1](#))
  - a. Locate the first rising edge of a “0” bit.
  - b. Count 32 bits.
  - c. Locate the rising edge of a “0” bit, this is the **start** of the measured time.
  - d. Count 32 bits.
  - e. Locate the rising edge of a “0” bit, this is the **end** of the measured time.
  - f. The reference bit length *fBitRateRef* equals 32 divided by the measured time.
8. The Tester finds a rising edge of a “0” bit and then it locates a 1024 bits long packet, where the first bit is identified as B0, and the last bit is identified as B1023. The Tester repeats the following 256 times with *\_num* = 0 (the first run starts from B0 through B32, the second runs start from B4 through B36 etc.)
  - a. Locate the rising edge of B(*\_num*) (a “0” bit), this is the **start** of the measured time.
  - b. Locate the rising edge of B(*\_num* + 32) (a “0” bit), this is the **end** of the measured time.
  - c. Calculate *fBitRateMeas* as 32 divided by the measured time.
  - d. The Tester checks *fBitRateMeas*: [\[TEST.PD.PHY.ALL.1#1\]](#)
    - i. The check fails if *fBitRateMeas* is less than or equal to *fBitRate* min.
    - ii. The check fails if *fBitRateMeas* is more than or equal to *fBitRate* max.
  - e. Compute  $pBitRateMeas = (|fBitRateMeas - fBitRateRef|) / fBitRateRef$ .
  - f. The Tester checks that *pBitRateMeas* is less than 0.25% [\[TEST.PD.PHY.ALL.1#2\]](#)
  - g. Set *\_num* = *\_num* + 4
9. The Tester checks that the bit stream collected meets *tBISTContMode* limits. [\[TEST.PD.PHY.ALL.1#3\]](#)
10. If it is a Cable Plug UUT, repeat the test on the other end of the cable.

### TEST.PD.PHY.ALL.2 Transmitter Eye Diagram

**Description:** The Tester checks that the UUT transmitted data meets the eye diagram mask requirements.

#### Test Specific Tester Behavior:

If bring-up will result in the UUT being a Sink:

- The Tester shall implement  $R_p$  as a  $4.7k\Omega$  resistor pull up to 3.3V. The capacitive load of the tester is as close to 400pF as practical.

If bring-up will result in the UUT being a Source:

- The Tester shall implement  $R_d$  as a  $5.1k\Omega$  resistor to ground. The capacitive load of the tester is as close to 400pF as practical.

If UUT is a Cable:

- The Tester applies  $V_{BUS}$ ,  $V_{CONN}$ ,  $R_p$  ( $4.7k\Omega$  to 3.3V) and  $R_d$  ( $5.1k\Omega$ ) to one cable end. The capacitive load of the tester is as close to 400pF as practical.

#### Test Conditions:

	Consumer Only	Provider Only, P/C	DRP, C/P	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3ChkdSrc</i>		✓			
<i>Rev3ChkdSnk</i>	✓		✓		
<i>CabFlippedSrc</i>		✓			
<i>CabFlippedSnk</i>	✓		✓		
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMidEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMidEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

#### Test Procedures:

1. There are 6 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider or DRP. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Provider Only or Provider/Consumer. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - c. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - d. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10

- e. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - f. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a BIST request to the UUT, specifying “BIST Carrier Mode”. If the UUT is acting as a Sink, do not follow the normal Rp Collision Avoidance but maintain the specified 4.7k $\Omega$  resistor pull up to 3.3V
3. The Tester creates a histogram with at least 100 bins of the total voltage swing for the bit stream collected from the BIST request. For example, 100 bins, the voltage absolute max is 1.1V and absolute min is 0V; then the top bin is count of voltages between 1.089V and 1.1V; bottom bin is count of voltages between 0V and 0.011V
4. The Tester searches for the bin with highest count between Y5TX (0.5625V) and the maximum; and the middle voltage of the bin is “high level”.
5. The Tester searches for the bin with highest count between Y5TX (0.5625V) and the minimum; and the middle voltage of the bin is “low level”.
6. The average between “high level” and “low level” is the mid-level.
7. Use the mid-level to find 1 UI (refer to [Figure 1](#))
  - a. Locate the rising edge of a "0" bit. This is the start of the measured time.
  - b. Count 32 bits.
  - c. Locate the rising edge of a "0" bit. This is the end of the measured time.
  - d. The reference bit length 1 UI equals the measured time divided by 32.
8. The Tester uses the mid-level as the initial crossing level of the Eye Diagram, refer to [Figure 2](#).
9. The Tester produces an Eye Diagram and checks that the parameters meet the requirements of Figures 5-22 and 5-23 of the USB PD2.0 specification.
  - a. The Eye Diagram shall use 2640 bits placed on the TX-Mask.
  - b. 1UI and crossing level is used for plotting the Eye Diagram. For each bit, the first and last crossings are separated equally from 0.5UI. Refer to [Figure 2](#).
  - c. The Eye Diagram check fails if the signals violate the TX-Mask.
10. If the Eye Diagram check fails in the previous step, the crossing level is shifted as follows and the Eye Diagram check is repeated for each sub-step below. For example, a given bit may shift left or right when the crossing level is adjusted.
  - a. Add 10mv to the initial crossing level to the mid-level, run Step-9
  - b. Subtract 10mv from the initial crossing level, run Step-9
  - c. Add 20mv to the initial crossing level, run Step-9
  - d. Subtract 20mv from the initial crossing level, run Step-9
  - e. Add 30mv to the initial crossing level, run Step-9
  - f. Subtract 30mv from the initial crossing level, run Step-9
  - g. Add 40mv to the initial crossing level, run Step-9
  - h. Subtract 40mv from the initial crossing level, run Step-9
  - i. Add 50mv to the initial crossing level, run Step-9

- j. Subtract 50mv from the initial crossing level, run Step-9
11. The check fails if the Eye Diagram check fails in Steps 9 and for all the crossing levels in 10. [TEST.PD.PHY.ALL.2#1]
12. The check fails if the rise time and fall time check fails. [TEST.PD.PHY.ALL.2#2]
  - a. A “10%-Level” is defined as “low level” + (“high level” - “low level”) \*10/100.
  - b. A “90%-Level” is defined as “low level” + (“high level” - “low level”) \*90/100.
  - c. For each rising edge of the 2640 bits, find the rise-time as the time from crossing the “10%-Level” until crossing the “90%-Level”. The final measured rise time is the average of these measurements. Check that this is not less than *tRise*.
  - d. For each falling edge of the 2640 bits, find the fall time as the time from crossing the “90%-Level” until crossing the “10%-Level”. The final measured fall time is the average of these measurements. Check that this is not less than *tFall*.
13. If the connected UUT is a port without a captive cable, repeat the test on the other CC connection by flipping the test cable plug at the UUT end.

### TEST.PD.PHY.ALL.3 Collision Avoidance

**Description:** The Tester checks that the UUT properly implements collision avoidance requirements.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, P/C	DRP, C/P	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3ChkdSrc</i>		✓			
<i>Rev3ChkdSnk</i>	✓		✓		
<i>Rev3eVcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

**Test Procedures:**

1. There are 6 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider or DRP. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Provider Only or Provider/Consumer. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - c. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - d. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10

- e. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
- f. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a BIST request to the UUT, specifying “BIST Test Data”.
3. The Tester waits *tCtsBIST*.
4. The Tester sends a BIST request to the UUT, specifying “BIST Test Data”.
5. Exactly 17us after the end of the previous packet, the Tester transmits a bit stream sequence (alternating 0s and 1s simulating a preamble) for a duration of 200 us. *Editorial note: it takes 2 UI (3.7us tUnitInterval max) to generate 3 transitions (nTransitionCount). So that is 25us (tInterFrameGap) minus 7.4us.*
6. The Tester waits for *tCtsGoodCRC* and the check fails if the UUT sends a *GoodCRC*. The check also fails if the Tester detects the tail end of the *GoodCRC* Message immediate after it has stopped transmitting in the previous step. [TEST.PD.PHY.ALL.3#1]
7. The Tester sends a BIST request to the UUT, specifying “BIST Test Data”.
8. Immediately after the end of the previous packet, the Tester transmit a continuous stream of 0s for a duration of 195us.
9. The Tester waits for *tCtsGoodCRC* and the check fails if the UUT sends a *GoodCRC*. The check also fails if the Tester detects the tail end of the *GoodCRC* Message immediate after it has stopped transmitting in the previous step. [TEST.PD.PHY.ALL.3#2]
10. If it is a Cable Plug UUT, repeat the test on the other end of the cable.

#### TEST.PD.PHY.ALL.4 Bus Idle Detection

**Description:** The Tester checks that the expected levels of noise on the CC line do not prevent the UUT from detecting a bus idle condition.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, P/C	DRP, C/P	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3ChkdSrc</i>		✓			
<i>Rev3ChkdSnk</i>	✓		✓		
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

**Test Procedures:**

1. There are 6 possible bring-up procedures:

- a. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider or DRP. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Provider Only or Provider/Consumer. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - c. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - d. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - f. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a BIST request to the UUT, specifying “BIST Test Data”.
  3. The Tester waits *tCtsBIST*.
  4. The Tester sends a BIST request to the UUT, specifying “BIST Test Data”.
  5. Immediately after the end of the previous packet, the Tester either transmits simulated noise waveforms using an Arbitrary Waveform Generator (AWG) or transmits the two-tone noise waveforms for 438us.
  6. The check fails if the UUT does not respond with a *GoodCRC* within *tReceive*.  
[TEST.PD.PHY.ALL.4#1]
  7. If it is a Cable Plug UUT, repeat the test on the other end of the cable.

The two-tone sinusoidal noise waveform is generated as followed:

1. The noise calibration setup is as shown in *Figure 3*:
  - a. The equivalent impedance of the Tester BMC transmitter and the noise generator is 50 Ohm +/- 10% and 400pF +/-15%
  - b. The calibration fixture consists of a bandwidth limiting filter as shown in *Figure 3*. The probe capacitance loading effect shall be considered.
2. The noise generator provides a superimposed two-tone sinusoidal waveform. One at 750 ± 50 kHz and another at 5 ± 0.5 MHz
3. The noise magnitude is calibrated using a scope connected to the calibration fixture (refer to *Figure 4*):
  - a. First, inject 750 kHz sinusoidal waveform and adjust the magnitude until the scope (attached to the calibration fixture) measures 90mV peak-to-peak.
  - b. Next, superimpose 5 MHz sinusoidal waveform on the 750 kHz waveform, adjust the magnitude of the 5 MHz sinusoidal waveform until the scope measures 300mV peak-to-peak.

The Arbitrary Waveform Generator (AWG) waveform is generated as followed:

1. The noise calibration setup is as shown in *Figure 3*:



- a. The equivalent impedance of the Tester BMC transmitter and the noise generator is 50 Ohm +/- 10% and 400pF +/-15%
  - b. The calibration fixture consists of a bandwidth limiting filter as shown in [Figure 3](#). The probe capacitance loading effect shall be considered.
2. Program the AWG to output the simulated noise using the generated waveform file. This should be summed with the CC signal during the transmission of a BIST Tester Data message.
3. In [Figure 5](#), the left-hand waveform m13 is both the content of the AWG-waveform data file, and exactly what should appear on the CC line of the test cable. The waveform data file is not what should be applied to the noise insertion point in [Figure 3](#). This point requires a pre-emphasized version of the waveform data file, which compensates for the filtering effect of the noise insertion circuit comprising the two 100Ω resistors, the 400pF and 200pF capacitors and the nominal capacitance of the cable type used by the test equipment.

**NOTE:** The timing of signal in presence of noise is shown in [Appendix F. Signal In Presence of Noise](#).

The noise is generated around the Y5TX Level.

#### *TEST.PD.PHY.ALL.5 Receiver Interference Rejection*

**Description:** The Tester checks that the UUT correctly receives signals under boundary conditions.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3ChkdSrc</i>		✓	✓		
<i>Rev3ChkdSnk</i>	✓		✓		
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

**Test Procedures:**

1. There are 9 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up

- procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
- c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a BIST request to the UUT, specifying “BIST Test Data”.
  3. The Tester waits *tCtsBISTtCtsWait*.
  4. Using TX\_GROUP\_1 parameters in the [Table 9](#) Receiver Test Signaling Table, the Tester sends “BIST Test Data” message (without incrementing *MessageID*) for 13362 times. The noise generator transmits two-tone noise or AWG waveforms when the Tester sends the BIST Tester Data message, and the two-tone noise or AWG waveforms are not transmitted when the Tester stops sending BIST Tester Data message. The check fails if UUT does not respond with a *GoodCRC* for each of the “BIST Test Data” message.  
[TEST.PD.PHY.ALL.5#1]
  5. The Tester runs the following procedures depending on the power role:
    - a. If it is a Sink UUT, using TX\_GROUP\_2 parameters in the [Table 9](#) Receiver Test Signaling Table, the Tester sends “BIST Test Data” message (without incrementing *MessageID*) for 13362 times. The check fails if UUT does not respond with a *GoodCRC* for each of the “BIST Test Data” message.  
[TEST.PD.PHY.ALL.5#2]



- b. If it is a Source UUT, using TX\_GROUP\_3 parameters in the [Table 9](#) Receiver Test Signaling Table, the Tester sends “BIST Test Data” message (without incrementing *MessageID*) for 13362 times. The check fails if UUT does not respond with a *GoodCRC* for each of the “BIST Test Data” message.  
[TEST.PD.PHY.ALL.5#3]
- c. If it is a Cable Plug UUT and “eMarker integrated on PD Controller = No”, the Tester performs the following checks:
  - i. Using TX\_GROUP\_2 parameters in the [Table 9](#) Receiver Test Signaling Table, the Tester sends “BIST Test Data” message (without incrementing *MessageID*) for 13362 times. The check fails if UUT does not respond with a *GoodCRC* for each of the “BIST Test Data” message  
[TEST.PD.PHY.ALL.5#4].
  - ii. Using TX\_GROUP\_3 parameters in the [Table 9](#) Receiver Test Signaling Table, the Tester sends “BIST Test Data” message (without incrementing *MessageID*) for 13362 times. The check fails if UUT does not respond with a *GoodCRC* for each of the “BIST Test Data” message  
[TEST.PD.PHY.ALL.5#5].
- d. If it is a Cable Plug UUT and “eMarker integrated on PD Controller = Yes” the Tester performs the following checks depends on the power role of DUT.
  - i. If it is a Sink UUT, using TX\_GROUP\_2 parameters in the [Table 9](#) Receiver Test Signaling Table, the Tester sends “BIST Test Data” message (without incrementing *MessageID*) for 13362 times. The check fails if the UUT does not respond with a *GoodCRC* for each of the “BIST Test Data” message [TEST.PD.PHY.ALL.5#6].
  - ii. Else, if it is a Source UUT, using TX\_GROUP\_3 parameters in the [Table 9](#) Receiver Test Signaling Table, the Tester sends “BIST Test Data” message (without incrementing *MessageID*) for 13362 times. The check fails if the UUT does not respond with a *GoodCRC* for each of the “BIST Test Data” message [TEST.PD.PHY.ALL.5#7].
8. If UUT is a Dual-Role Power device (and it has been running this test first time as a Sink), repeat the test with the UUT as a Source. If it is a Cable Plug UUT, repeat the test on the other end of the cable.

The two-tone sinusoidal noise waveform is generated as followed:

1. The noise calibration setup is as shown in [Figure 3](#):
  - a. The equivalent impedance of the Tester BMC transmitter and the noise generator is 50 Ohm +/- 10% and 400pF +/-15%
  - b. The calibration fixture consists of a bandwidth limiting filter as shown in [Figure 3](#). The probe capacitance loading effect shall be considered.
2. The noise generator provides a superimposed two-tone sinusoidal waveform. One at  $750 \pm 50$  kHz and another at  $5 \pm 0.5$  MHz
3. The noise magnitude is calibrated using a scope connected to the calibration fixture (refer to [Figure 4](#)):
  - a. First, inject 750 kHz sinusoidal waveform and adjust the magnitude until the scope (attached to the calibration fixture) measures 90mV peak-to-peak. The offset is at 0V.

- b. Next, superimpose 5 MHz sinusoidal waveform on the 750 kHz waveform, adjust the magnitude of the 5 MHz sinusoidal waveform until the scope measures 165mV peak-to-peak. The offset is at 0V.

The Arbitrary Waveform Generator (AWG) waveform is generated as followed:

1. The noise calibration setup is as shown in [Figure 3](#):
  - a. The equivalent impedance of the Tester BMC transmitter and the noise generator is 50 Ohm +/- 10% and 400pF +/-15%
  - b. The calibration fixture consists of a bandwidth limiting filter as shown in [Figure 3](#). The probe capacitance loading effect shall be considered.
2. Program the AWG to output the simulated noise using the generated waveform file. This should be summed with the CC signal during the transmission of a BIST Tester Data message.
3. In [Figure 5](#), the left-hand waveform m13 is both the content of the AWG-waveform data file, and exactly what should appear on the CC line of the test cable. The waveform data file is not what should be applied to the noise insertion point in [Figure 3](#). This point requires a pre-emphasized version of the waveform data file, which compensates for the filtering effect of the noise insertion circuit comprising the two 100Ω resistors, the 400pF and 200pF capacitors and the nominal capacitance of the cable type used by the test equipment.

**NOTE:** The timing of signal in presence of noise is shown in [Appendix F. Signal In Presence of Noise](#).

The noise is generated around the Y5TX Level.

#### *TEST.PD.PHY.ALL.6 Invalid SOP\**

**Description:** The Tester checks that the UUT responds correctly to invalid SOP\*.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, P/C	DRP, C/P	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3ChkdSrc</i>		✓			
<i>Rev3ChkdSnk</i>	✓		✓		
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

**Test Procedures:**

1. There are 6 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider or DRP. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2

- b. The UUT has VIF field PD\_Port\_Type set to Provider Only or Provider/Consumer. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - c. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - e. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - f. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - g. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a BIST request to the UUT, specifying “BIST Test Data”.
  3. The Tester waits *tCtsBIST* to respond to messages from the UUT.
  4. Transmit a BIST message with BIST Test Data object using SOP packet.
  5. Transmit 10 BIST Test Data messages flipping Bit 0 of the first two symbols (number 1&2) of the SOP. The check fails if the UUT responds with a *GoodCRC*.  
[TEST.PD.PHY.ALL.6#1]
  6. Repeat Step-5 with all combinations of flipping Bit 0 of two symbols of the SOP (number 1&3, 1&4, 2&3, 2&4, 3&4).
  7. Additionally, for Cable Plug, repeat the test using SOP’ packet (if SOP\_P\_Capable = Yes) and SOP’’ packet (if SOP\_PP\_Capable = Yes).

#### TEST.PD.PHY.ALL.7 Valid SOP\*

**Description:** The Tester checks that the UUT responds correctly to valid SOP\*.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, P/C	DRP, C/P	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3ChkdSrc</i>		✓			
<i>Rev3ChkdSnk</i>	✓		✓		
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

### Test Procedures:

1. There are 6 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider or DRP. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Provider Only or Provider/Consumer. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - c. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - d. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - f. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends 10 BIST Test Data messages using SOP packet.
  - a. If SOP\_Capable = NO, the check fails if the UUT respond with a *GoodCRC* [TEST.PD.PHY.ALL.7#1]
  - b. If SOP\_Capable = YES, the check fails if the UUT does not respond with a *GoodCRC*. [TEST.PD.PHY.ALL.7#2]
3. The Tester sends 10 BIST Test Data messages using SOP'.
  - a. If SOP\_P\_Capable = NO, the check fails if the UUT responds with a *GoodCRC*. [TEST.PD.PHY.ALL.7#3]
  - b. The check fails if
    - i. VIF\_Product\_Type = Port **and**
    - ii. PD\_Port\_Type != eMarker **and**
    - iii. Captive\_Cable\_Is\_eMarked = NO **and**
    - iv. UUT is not the Vconn Sourceand UUT does respond with a *GoodCRC*. [TEST.PD.PHY.ALL.7#9]
  - c. The check fails if
    - i. VIF\_Product\_Type = Cable or
    - ii. PD\_Port\_Type = eMarker or
    - iii. Captive\_Cable\_Is\_eMarked = YESthe UUT does not respond with a *GoodCRC* [TEST.PD.PHY.ALL.7#4]
4. The Tester sends 10 BIST Test Data messages using SOP''.
  - a. If SOP\_PP\_Capable = NO, the check fails if the UUT responds with a *GoodCRC*. [TEST.PD.PHY.ALL.7#5]
  - b. The check fails if
    - i. VIF\_Product\_Type = Port **and**

- ii. PD\_Port\_Type != eMarker **and**
  - iii. Captive\_Cable\_Is\_eMarked = NO **and**
  - iv. UUT is not the Vconn Source
 and UUT responds with a *GoodCRC*. [TEST.PD.PHY.ALL.7#10]
- c. The check fails if
  - i. (VIF\_Product\_Type = Cable or Captive\_Cable\_Is\_eMarked = YES) and
  - ii. SOP\_PP\_Capable = YES
 and the UUT does not respond with a *GoodCRC*. [TEST.PD.PHY.ALL.7#6]
- 5. The Tester sends 10 BIST Test Data messages using SOP' \_Debug.
  - a. The check fails if SOP\_P\_Debug\_Capable = NO and the UUT responds with a *GoodCRC* [TEST.PD.PHY.ALL.7#7]
- 6. The Tester sends 10 BIST Test Data messages using SOP'' \_Debug.
  - a. The check fails if SOP\_PP\_Debug\_Capable = NO and the UUT responds with a *GoodCRC* [TEST.PD.PHY.ALL.7#8]

#### TEST.PD.PHY.ALL.8 Incorrect CRC

**Description:** The Tester checks that the UUT does not provides a *GoodCRC* to an intentionally corrupted message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, P/C	DRP, C/P	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3ChkdSrc</i>		✓			
<i>Rev3ChkdSnk</i>	✓		✓		
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

#### Test Procedures:

1. There are 7 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider or DRP. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Provider Only or Provider/Consumer. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - c. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - d. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10

- e. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
- f. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a message depending on the UUT type/role (note that this test is first run sending uncorrupted message followed by corrupted messages):
  - a. If the UUT is a Cable Plug, the Tester sends a *Discover Identity* Command.
  - b. If the UUT is a Source or Sink, the Tester sends a *Get\_Sink\_Cap* Message.
3. The Tester checks the response message depending on the UUT type/role:
  - a. The UUT is a Cable Plug: [TEST.PD.PHY.ALL.8#1]
    - i. If the Tester has sent an uncorrupted message, the check fails if the UUT does not send a *Discover Identity* ACK message.
    - ii. If the Tester has sent a corrupted message, the check fails if the UUT responds with a *GoodCRC* or a *Discover Identity* ACK message.
  - b. The UUT is a Sink: [TEST.PD.PHY.ALL.8#2]
    - i. If the Tester has sent an uncorrupted message, the check fails if the UUT does not send a *Sink\_Capabilities* Message.
    - ii. If the Tester has sent a corrupted message, the check fails if the UUT responds with a *GoodCRC* or a *Sink\_Capabilities* Message.
  - c. The UUT is a Source: [TEST.PD.PHY.ALL.8#3]
    - i. If the Tester has sent an uncorrupted message, the check fails if the UUT does not send *Sink\_Capabilities*, *Not\_Supported* or *Reject* Message.
    - ii. If the Tester has sent a corrupted message, the check fails if the UUT responds with a *GoodCRC* or *Sink\_Capabilities*, *Not\_Supported* or *Reject* Message.
4. The Tester repeats Step-2 and Step-3 but implements modifications to the messages in Step-2 as followed:
  - a. Flip Bit 0 in the CRC before 4b5b encoding.
  - b. Flip Bit 0 in the CRC after 4b5b encoding.
  - c. After generating the CRC, flip Bit 0 of the packet payload before 4b5b encoding
  - d. After generating the CRC and the 4b5b encoding, flip Bit 0 of the packet payload
  - e. After the 4b5b encoding, replace the third 5b symbol of the Message Header with 11111b (a reserved 5b symbol).

#### *TEST.PD.PHY.ALL.9 Receiver Input Impedance*

**Description:** The Tester checks that the UUT receiver input impedance is not excessively low. It also checks that UUT does not modify voltage levels in any received message waveform. It is not practical to directly measure the input impedance of the receiver in the UUT (required to be  $\geq 1 \text{ M}\Omega$ ), as it is masked by the  $R_p / R_d$  resistors. Instead, we will assume the presence of these resistors and measure that the resulting resistance falls within a valid range.

**Test Specific Tester Behavior:** N/A



#### Test Conditions:

	Consumer Only	Provider Only, P/C	DRP, C/P	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3ChkdSrc</i>		✓			
<i>Rev3ChkdSnk</i>	✓		✓		
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

#### Test Procedures:

9. There are 6 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider or DRP. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Provider Only or Provider/Consumer. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - c. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - d. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - f. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
10. The Tester sends a BIST request to the UUT, specifying “BIST Test Data”.
11. Depending on the UUT
  - a. For Sink: Check that the voltage on the CC line falls within the expected range, knowing the values of the Rp and Rd resistors [TEST.PD.PHY.ALL.9#1]
  - b. For Source: Check that the voltage on the CC line falls within one of the two expected ranges, knowing the possible values of the Rp and Rd resistors [TEST.PD.PHY.ALL.9#2]
  - c. For Cable Plug: Check that the voltage on the CC line falls within the expected range, knowing the value of these resistors [TEST.PD.PHY.ALL.9#3]

12. Reset UUT by simulating a tester end cable detach, in order to guarantee exiting the BIST Mode.
13. Depending on the UUT
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider or DRP. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Provider Only or Provider/Consumer. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - c. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
14. The Tester sends a BIST request to the UUT, specifying “BIST Test Data”.
15. Check that signal voltages on the CC line of -300mV and 1500mV are not modified at the receiver input by more than the tester tolerances may produce. The actual test method is left to the discretion of the implementer. [TEST.PD.PHY.ALL.9#4]  
Except for cable plug, end test here.
16. For Cable Plug only:  
The Tester applies Rp (4.7kΩ to 3.3V) and Rd (5.1kΩ to 0V), but not VCONN or VBUS to one cable end.
17. Check that signal voltages on the CC line of -300mV and 1500mV are not modified at the receiver input by more than the tester tolerances may produce. The actual test method is left to the discretion of the implementer. [TEST.PD.PHY.ALL.9#5]
18. Repeat last two steps at other end of Cable.

## 4.2 Test Procedures for Source and Sink Capable UUTs

### 4.2.1 PD3 Mode

The tests in this subsection are run only in PD3 mode.

The tests in this subsection are applicable to UUT with VIF field VIF\_Product\_Type set to Port.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (ie, a Provider/Consumer or Consumer/Provider)
- The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### TEST.PD.PHY.PORT.1 Invalid Reset Signals

Description: The Tester checks that the UUT responds correctly to invalid reset signals. Also, the Tester checks that the UUT ignores messages upon reception of a BIST Test Data BIST data Object.



Test Specific Tester Behavior: N/A

	Consumer Only	Provider Only	DRP, C/P, P/C
<i>Rev3ChkdSrc</i>		✓	✓
<i>Rev3ChkdSnk</i>	✓		✓

Test Procedures:

- There are 5 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
- The Tester sends a BIST request to the UUT, specifying “BIST Test Data”.
- The Tester waits *tCtsBIST* to respond to messages from the UUT.
- The Tester sends a *Cable Reset* Signal.
- The Tester waits 1.5 second, and the check fails if the UUT sends any message or removes  $V_{BUS}$ . [TEST.PD.PHY.PORT.1#1]
- The Tester transmits *Hard Reset* flipping Bit 0 of the first two symbols (number 1&2).
- The Tester waits 1.5 second and checks that the UUT does not send any message or removing  $V_{BUS}$ .
- The Tester sends a *Get\_Sink\_Cap* Message to the UUT.
- The Tester waits for *tCtsSenderResponse* (see **Table 19** Timing Table & Calculations), and the check fails if the UUT sends a response to *Get\_Sink\_Cap* Message. [TEST.PD.PHY.PORT.1#2]
- The Tester repeats Step-6 through Step-9 and uses all combinations of flipping Bit 0 of two symbols (number 1&3, 1&4, 2&3, 2&4, 3&4) in Step-6.



## 5 Protocol Specific Tests

This chapter describes tests that primary cover Chapters 6 and 8 requirements in the USB PD2 specifications.

### 5.1 Test Procedures for All UUTs

#### 5.1.1 PD2 and PD3 Modes

The tests in this subsection are run firstly in PD2 mode and secondly in PD3 mode.

The tests in this subsection are applicable to all UUTs.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (ie, a Provider/Consumer or Consumer/Provider)
- The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### TEST.PD.PROT.ALL.1 Corrupted GoodCRC

Description: The Tester checks that the UUT retransmits the PD message if it does not receive a correct *GoodCRC*.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
Rev2Src		✓	✓		
Rev3ChkdSrc		✓	✓		
Rev2Snk	✓		✓		
Rev3ChkdSnk	✓		✓		
Rev2VcMinEnd1				✓	✓
Rev3VcMinEnd1				✓	✓
Rev2VcMaxEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev2VcMinEnd2				✓	
Rev3VcMinEnd2				✓	
Rev2VcMaxEnd2				✓	
Rev3VcMaxEnd2				✓	

Test Procedures:

1. There are 9 possible bring-up procedures:

- a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a message depending on the UUT type/role (first time):
    - a. If the UUT is a Cable Plug, the Tester sends a *Discover Identity* Command.
    - b. If the UUT is a Source or Sink, the Tester sends a *Get\_Sink\_Cap* Message.
  3. The Tester checks the response message depending on the UUT type/role:
    - a. If the UUT is a Cable Plug, the check fails if the UUT does not send a *Discover Identity* ACK message. [TEST.PD.PROT.ALL.1#1]
    - b. If the UUT is a Sink, the check fails if the UUT does not send a *Sink\_Capabilities* Message. [TEST.PD.PROT.ALL.1#2]

- c. If the UUT is a Source, the Tester checks the response as follows:  
 [TEST.PD.PROT.ALL.1#3]
      - i. If the VIF field PD\_Port\_Type is set to Provider/Consumer, check that the UUT sends a *Sink\_Capabilities* Message.
      - ii. In PD2 mode, if the VIF field PD\_Port\_Type is set to Provider Only, check that the UUT sends a *Reject* Message.
      - iii. In PD3 mode, if the VIF field PD\_Port\_Type is set to Provider Only, check that the UUT sends a *Not\_Supported* Message.
  4. The Tester sends a *GoodCRC* with an incorrect CRC (flip bit 0 of the CRC before 4b5b encoding).
  5. The Tester checks the behavior:
    - a. For Cable Plug UUT, the check fails if the UUT retransmits  
 [TEST.PD.PROT.ALL.1#4]
    - b. For Sink or Source UUT, the check fails if the UUT does not retransmit (*Sink\_Capabilities*, *Not\_Supported* or *Reject* Message) with the same *MessageID* (within *tRetry* of the *CRCReceiveTimer* expiring), and the Tester responds with a valid *GoodCRC*. [TEST.PD.PROT.ALL.1#5]
  6. The Tester sends a message depending on the UUT type/role (second time):
    - a. If the UUT is a Cable Plug, the Tester sends a *Discover Identity* Command.
    - b. If the UUT is a Source or Sink, the Tester sends a *Get\_Sink\_Cap* Message.
  7. The Tester checks the response message depending on the UUT type/role:
    - a. If the UUT is a Cable Plug, the check fails if the UUT does not send a *Discover Identity* ACK message. [TEST.PD.PROT.ALL.1#6]
    - b. If the UUT is a Sink, the check fails if the UUT does not send a *Sink\_Capabilities* Message. [TEST.PD.PROT.ALL.1#7]
    - c. If the UUT is a Source, the check fails if the UUT does not send *Sink\_Capabilities*, *Not\_Supported* or *Reject* Message.  
 [TEST.PD.PROT.ALL.1#8]
  8. The Tester sends a *GoodCRC* with an incorrect *MessageID* (increment by one from the received message).
  9. The Tester checks the behavior:
    - a. For Cable Plug UUT, the check fails if the UUT retransmits  
 [TEST.PD.PROT.ALL.1#9]
    - b. For Sink or Source UUT, the check fails if the UUT does not retransmit (*Sink\_Capabilities*, *Not\_Supported* or *Reject* Message) with the same *MessageID* (within *tRetry* of the *CRCReceiveTimer* expiring). [TEST.PD.PROT.ALL.1#10]
  10. For Cable Plug UUT, the test ends here. For Sink or Source UUT, the Tester repeats the previous 2 steps until the UUT has retransmitted the same *MessageID* for nRetryCount times.
  11. For a Cable Plug the test ends here.
- ## The following tests if the UUT port discards an outgoing message when receiving an incoming message.
- ## Figure 6-61 “Common Protocol Layer Message Transmission State Diagram”
- ## entry into PRL\_Tx\_Discard\_Message state
12. The Tester sends a *Get\_Sink\_Cap* Message. (beginning of the AMS)
  13. The Tester waits for the *Sink\_Capabilities*, *Not\_Supported* or *Reject* Message. (e

14. The Tester sends a *GoodCRC* with an incorrect CRC (flip bit 0 of the CRC before 4b5b encoding).
15. The Tester sends a *Get\_Source\_Cap* Message 35μs (before *tRetry* expires) after the incorrect the *GoodCRC* Message has been sent. (The Tester sends another AMS while the first one is ongoing)
16. The check fails if the UUT retries the *Sink\_Capabilities*, *Not\_Supported* or *Reject* Message with the same MessageID after receiving the *Get\_Source\_Cap* Message. The Tester waits 25ms for the retry. [TEST.PD.PROT.ALL.1#11]
17. The check fails if the UUT does not send a *Soft\_Reset* Message. (*Soft\_Reset* is expected if an AMS got interrupted) [TEST.PD.PROT.ALL.1#12]

#### TEST.PD.PROT.ALL.2 Soft Reset and Hard Reset

**Description:** The Tester checks that the UUT uses *Soft\_Reset* and *Hard\_Reset* correctly.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
Rev2Src		✓	✓		
Rev3ChkdSrc		✓	✓		
Rev2Snk	✓		✓		
Rev3ChkdSnk	✓		✓		
Rev2VcMinEnd1				✓	✓
Rev3VcMinEnd1				✓	✓
Rev2VcMaxEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev2VcMinEnd2				✓	
Rev3VcMinEnd2				✓	
Rev2VcMaxEnd2				✓	
Rev3VcMaxEnd2				✓	

#### Test Procedures:

1. There are 9 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).

- c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a message depending on the UUT type/role (first time):
    - a. If the UUT is a Cable Plug, the Tester sends a *Discover Identity* Command.
    - b. If the UUT is a Source or Sink, the Tester sends a *Get\_Sink\_Cap* Message.
  3. The Tester checks the response message depending on the UUT type/role:
    - a. If the UUT is a Cable Plug, the check fails if the UUT does not send a *Discover Identity* ACK message. [TEST.PD.PROT.ALL.2#1]
    - b. If the UUT is a Sink, the check fails if the UUT does not send a *Sink\_Capabilities* Message. [TEST.PD.PROT.ALL.2#2]
    - c. If the UUT is a Source, the Tester checks the response as follows: [TEST.PD.PROT.ALL.2#3]
      - i. If the VIF field PD\_Port\_Type is set to Provider/Consumer, check that the UUT sends a *Sink\_Capabilities* Message.
      - ii. In PD2 mode, if the VIF field PD\_Port\_Type is set to Provider Only, check that the UUT sends a *Reject* Message.
      - iii. In PD3 mode, if the VIF field PD\_Port\_Type is set to Provider Only, check that the UUT sends a *Not\_Supported* Message.
  4. The Tester does not send a *GoodCRC*.



5. The Tester checks the response message depending on the UUT type/role:
  - a. For Cable Plug UUT, the check fails if the UUT retransmits  
[TEST.PD.PROT.ALL.2#4]
  - b. For Sink or Source UUT, the check fails if the UUT does not retransmit  
(*Sink\_Capabilities*, *Not\_Supported* or *Reject* Message) with the same *MessageID*  
(within *tRetry* of the *CRCReceiveTimer* expiring) for *nRetryCount* times.  
[TEST.PD.PROT.ALL.2#5]
6. The Tester checks the behavior depending on the UUT type/role:
  - a. If the UUT is a Cable Plug, the Tester waits 35ms and the check fails if the UUT  
sends a *Soft Reset* Message or a *Hard Reset* Signal. [TEST.PD.PROT.ALL.2#6]
  - b. If the UUT is a Sink
    - i. The Tester checks if the UUT transmits a *Soft Reset* Message within  
*tSoftReset* of *tReceive* expiring. The check fails if the *Soft Reset* Message  
is not received within the following time interval [*tReceive* min, *tReceive*  
max + *tSoftReset* max] or [0.9ms, 16.1ms]. [TEST.PD.PROT.ALL.2#7]  
The Tester responds with a valid *GoodCRC*.
    - ii. The Tester sends an *Accept* Message.
    - iii. The Tester transmits *Source\_Capabilities* Message with single PDO (5V  
@ 500mA, Source only). The check fails if the UUT does not send a  
*Request* Message. [TEST.PD.PROT.ALL.2#8] The Tester establishes a  
PD contract using common procedure COMMON.PROC.PD.10.
  - c. If the UUT is a Source
    - i. The Tester checks if the UUT transmits a *Soft Reset* Message within  
*tSoftReset* of *tReceive* expiring. The check fails if the *Soft Reset* Message  
is not received within the following time interval [*tReceive* min, *tReceive*  
max + *tSoftReset* max] or [0.9ms, 16.1ms]. [TEST.PD.PROT.ALL.2#9]  
The Tester responds with a valid *GoodCRC*.
    - ii. The Tester sends an *Accept* Message.
    - iii. The check fails if the UUT does not send a *Source\_Capabilities* Message.  
[TEST.PD.PROT.ALL.2#10] The Tester establishes a PD contract using  
common procedure COMMON.PROC.PD.11.
7. The Tester sends a message depending on the UUT type/role (second time):
  - a. If the UUT is a Cable Plug, the Tester sends a *Discover Identity* Command.
  - b. If the UUT is a Source or Sink, the Tester sends a *Get\_Sink\_Cap* Message.
8. The Tester checks the behavior depending on the UUT type/role:
  - a. For Cable Plug UUT: the check fails if the UUT does not send a *Discover Identity*  
ACK message. [TEST.PD.PROT.ALL.2#11]
  - b. For Sink UUT: the check fails if the UUT does not send a *Sink\_Capabilities*  
Message. [TEST.PD.PROT.ALL.2#12]
  - c. For Source UUT: the check fails if the UUT does not send *Sink\_Capabilities*,  
*Not\_Supported* or *Reject* Message. [TEST.PD.PROT.ALL.2#13]
9. The Tester does not send a *GoodCRC*.
10. The Tester checks the response message depending on the UUT type/role:
  - a. For Cable Plug UUT, the check fails if the UUT retransmits  
[TEST.PD.PROT.ALL.2#14]



- b. For Sink or Source UUT, the check fails if the UUT does not retransmit (*Sink\_Capabilities*, *Not\_Supported* or *Reject* Message) with the same *MessageID* (within *tRetry* of the *CRCReceiveTimer* expiring) for *nRetryCount* times. [TEST.PD.PROT.ALL.2#15]
- 11. The Tester checks the behavior depending on the UUT type/role:
  - a. If the UUT is a Cable Plug, check that the UUT does not send a *Soft Reset* Message or a *Hard Reset* Signal [TEST.PD.PROT.ALL.2#16]
  - b. If the UUT is a Sink or Source
    - i. The Tester checks if the UUT transmits a *Soft Reset* Message within *tSoftReset* of *tReceive* expiring. The check fails if the *Soft Reset* Message is not received within the following time interval [*tReceive* min, *tReceive* max + *tSoftReset* max] or [0.9ms, 16.1ms]. [TEST.PD.PROT.ALL.2#17]
    - ii. The Tester does not send a *GoodCRC* to the *Soft Reset* Message.
    - iii. The check fails if the UUT does not retransmit (*Soft Reset* Message) with the same *MessageID* (within *tRetry* of the *CRCReceiveTimer* expiring) for *nRetryCount* times [TEST.PD.PROT.ALL.2#18].
    - iv. The check fails if the UUT does not transmit a *Hard Reset* within *tHardReset* + *tReceive* max after *nRetryCount* of *Soft Reset* Message have failed [TEST.PD.PROT.ALL.2#19].

#### TEST.PD.PROT.ALL.3 Soft Reset Response

**Description:** The Tester checks that the UUT responds correctly to *Soft Reset* Message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
Rev2Src		✓	✓		
Rev3ChkdSrc		✓	✓		
Rev2Snk	✓		✓		
Rev3ChkdSnk	✓		✓		
Rev2VcMinEnd1				✓	✓
Rev3VcMinEnd1				✓	✓
Rev2VcMaxEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev2VcMinEnd2				✓	
Rev3VcMinEnd2				✓	
Rev2VcMaxEnd2				✓	
Rev3VcMaxEnd2				✓	

**Test Procedures:**

1. There are 9 possible bring-up procedures:

- a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a message depending on the UUT type/role (using an incremented *MessageID*):
    - a. If the UUT is a Cable Plug, the Tester sends a *Discover Identity* Command.
    - b. If the UUT is a Source or Sink, the Tester sends a *Get\_Sink\_Cap* Message.
  3. The Tester checks the response message (with an incremented *MessageID*) based on the UUT type/role:
    - a. If the UUT is a Cable Plug, the check fails if the UUT does not send a *Discover Identity* ACK message. [TEST.PD.PROT.ALL.3#1]

- b. If the UUT is a Sink, the check fails if the UUT does not send a *Sink\_Capabilities* Message. [TEST.PD.PROT.ALL.3#2]
  - c. If the UUT is a Source, the Tester checks the response as follows: [TEST.PD.PROT.ALL.3#3]
    - i. If the VIF field PD\_Port\_Type is set to Provider/Consumer, check that the UUT sends a *Sink\_Capabilities* Message.
    - ii. In PD2 mode, if the VIF field PD\_Port\_Type is set to Provider only, check that the UUT sends a *Reject* Message.
  - d. In PD3 mode, if the VIF field PD\_Port\_Type is set to Provider only, check that the UUT sends a *Not\_Supported* Message.
4. If *MessageID* in the last sent message is not 000b, the Tester repeats the previous 2 steps (i.e. sending *Discover Identity* Command or *Get\_Sink\_Cap* and receiving response) until the *MessageID* in the last sent message is 000b.
5. Immediately after sending *GoodCRC* (to the *Discover Identity* ACK, *Sink\_Capabilities* or *Reject*), the Tester sends a *Soft Reset* Message after *InterframeGap*. The check fails if the UUT does not send an *Accept* Message with *MessageID* 000b. [TEST.PD.PROT.ALL.3#4]
6. The Tester checks the response message depending on the UUT type/role:
  - a. For Sink UUT, the Tester transmits *Source\_Capabilities* Message with single PDO (5V @ 500mA, Source only). The check fails if the UUT does not send a *Request* Message. [TEST.PD.PROT.ALL.3#5] The Tester establishes a PD using common procedure COMMON.PROC.PD.10.
  - b. For Source UUT, the check fails if the UUT does not send a *Source\_Capabilities* Message within tFirstSourceCap after the last bit of the *GoodCRC* Message EOP corresponding to the *Accept* Message. [TEST.PD.PROT.ALL.3#6] Once the Tester receives *Source\_Capabilities* Message, the Tester establishes a PD contract using common procedure COMMON.PROC.PD.11.
7. The Tester sends a message depending on the UUT type/role:
  - a. If the UUT is a Cable Plug, the Tester sends a *Discover Identity* Command.
  - b. If the UUT is a Source or Sink, the Tester sends a *Get\_Sink\_Cap* Message.
8. The Tester checks the response message depending on the UUT type/role:
  - a. For Cable Plug UUT: the check fails if the UUT does not send a *Discover Identity* ACK message. [TEST.PD.PROT.ALL.3#7]
  - b. For Sink UUT: the check fails if the UUT does not send a *Sink\_Capabilities* Message. [TEST.PD.PROT.ALL.3#8]
  - c. For Source UUT: the check fails if the UUT does not send *Sink\_Capabilities*, *Not\_Supported* or *Reject* Message. [TEST.PD.PROT.ALL.3#9]

#### TEST.PD.PROT.ALL.4 Reset Signals and MessageID

**Description:** The Tester checks that the UUT responds correctly to *Hard Reset* and *Cable Reset* Signaling.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-
---------------	---------------	---------------	------------	-------------------------

					Only eMarker
Rev2Src		✓	✓		
Rev3ChkdSrc		✓	✓		
Rev2Snk	✓		✓		
Rev3ChkdSnk	✓		✓		
Rev2VcMinEnd1				✓	✓
Rev3VcMinEnd1				✓	✓
Rev2VcMaxEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev2VcMinEnd2				✓	
Rev3VcMinEnd2				✓	
Rev2VcMaxEnd2				✓	
Rev3VcMaxEnd2				✓	

#### Test Procedures:

1. There are 9 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3

- g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
- h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
- i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a message depending on the UUT type/role (using an incremented *MessageID*):
  - a. If the UUT is a Cable Plug, the Tester sends a *Discover Identity* Command.
  - b. If the UUT is a Source or Sink, the Tester sends a *Get\_Sink\_Cap* Message.
3. The Tester checks the response message (with an incremented *MessageID*) based on the UUT type/role:
  - a. If the UUT is a Cable Plug, check that the UUT sends a *Discover Identity* ACK message. [TEST.PD.PROT.ALL.4#1]
  - b. If the UUT is a Sink, check that the UUT sends a *Sink\_Capabilities* Message. [TEST.PD.PROT.ALL.4#2]
  - c. If the UUT is a Source, the Tester checks the response as follows: [TEST.PD.PROT.ALL.4#3]
    - i. If the VIF field PD\_Port\_Type is set to Provider/Consumer or DRP, check that the UUT sends a *Sink\_Capabilities* Message.
    - ii. In PD2 mode, if the VIF field PD\_Port\_Type is set to Provider only, check that the UUT sends a *Reject* Message.
    - iii. In PD3 mode, if the VIF field PD\_Port\_Type is set to Provider only, check that the UUT sends a *Not\_Supported* Message.
4. If *MessageID* in the last sent message is not 000b, the Tester repeats the previous 2 steps (i.e. sending *Discover Identity* REQ or *Get\_Sink\_Cap* and receiving response) until the *MessageID* in the last sent message is 000b.
5. Immediately after sending *GoodCRC* (to the *Discover Identity* ACK, *Sink\_Capabilities* or *Reject*), the Tester sends a Hard-Reset signal after *InterframeGap* (note that the second run through is a *Cable Reset* Signal for a Cable Plug UUT).
6. The test proceeds depending on the UUT type/role:
  - a. If the UUT is a Sink, the Tester drives  $V_{BUS}$  to *vSafe0V* and then restores  $V_{BUS}$  to *vSafe5V*. The Tester transmits *Source\_Capabilities* Messages with single PDO (5V @ 500mA, Source only) immediately after  $V_{BUS}$  reaches *vSafe0V* min. The Tester checks that the UUT sends a *Request* Message. [TEST.PD.PROT.ALL.4#4] Once it receives a *Request* Message, it proceeds using common procedure COMMON.PROC.PD.10. The Tester then waits *tCtsAfterLastPoint* to respond to messages from the UUT.
  - b. If the UUT is a Source:

- i. The Tester checks that the UUT drives  $V_{BUS}$  to *vSafe0V* and then restores  $V_{BUS}$  to *vSafe5V*. [TEST.PD.PROT.ALL.4#5]
  - ii. The check fails if *Source\_Capabilities* is not sent by the UUT within *tFirstSourceCap*. [TEST.PD.PROT.ALL.4#6] Once the Tester receives *Source\_Capabilities* Message, the Tester establishes a PD contract using common procedure COMMON.PROC.PD.11. The Tester then waits *tCtsAfterLastPoint* to respond to messages from the UUT.
- c. If the UUT is a Cable Plug, the Tester checks the response depending on the **run number**. [TEST.PD.PROT.ALL.4#7]
  - i. If the Tester has sent a *Hard Reset* (**first run**), the Tester drives  $V_{BUS}$  to *vSafe0V* and then restores  $V_{BUS}$  to *vSafe5V*. The Tester run VCONN power cycles process. The Tester transmits *Discover Identity* REQ every *tDiscoverIdentity* until either it receives a *GoodCRC* followed by a *Discover Identity* ACK or it has transmitted *Discover Identity* REQ 3 times. The check fails when the Tester has transmitted *Discover Identity* REQ for 3 times.
  - ii. If the Tester has sent a *Cable Reset* (**second run**), the Tester transmits *Discover Identity* REQ every *tDiscoverIdentity*. The check fails if the UUT does not sends a *GoodCRC* followed by a *Discover Identity* ACK.
- 7. The Tester sends a message depending on the UUT type/role (using an incremented *MessageID*):
  - a. If the UUT is a Cable Plug, the Tester sends a *Discover Identity* Command.
  - b. If the UUT is a Source or Sink, the Tester sends a *Get\_Sink\_Cap* Message.
- 8. The Tester checks the response message (with an incremented *MessageID*) based on the UUT type/role:
  - a. If the UUT is a Cable Plug, check that the UUT sends a *Discover Identity* ACK message. [TEST.PD.PROT.ALL.4#8]
  - b. If the UUT is a Sink, check that the UUT sends a *Sink\_Capabilities* Message. [TEST.PD.PROT.ALL.4#9]
  - c. If the UUT is a Source, the Tester checks the response as follows: [TEST.PD.PROT.ALL.4#10]
    - i. If the VIF field PD\_Port\_Type is set to Provider/Consumer, check that the UUT sends a *Sink\_Capabilities* Message.
    - ii. In PD2 mode, if the VIF field PD\_Port\_Type is set to Provider Only, check that the UUT sends a *Reject* Message.
    - iii. In PD3 mode, if the VIF field PD\_Port\_Type is set to Provider Only, check that the UUT sends a *Not\_Supported* Message.
- 9. The Tester repeats Step-7 with the same *MessageID*. Because the UUT is expected to ignore these erroneous messages, the Tester should immediately return to PE\_SRC\_Ready/PE\_SNK\_Ready after transmission.
- 10. The check fails if the UUT has sent a response message (either *Discover Identity* ACK, NAK or BUSY, or *Sink\_Capabilities* or *Reject*) when the *MessageID* has been repeated. [TEST.PD.PROT.ALL.4#11]
- 11. The Tester sends a message depending on the UUT type/role (using an incremented *MessageID*):
  - a. If the UUT is a Cable Plug, the Tester sends a *Discover Identity* Command.



- b. If the UUT is a Source or Sink, the Tester sends a *Get\_Sink\_Cap* Message.
12. The Tester checks the response message (with an incremented *MessageID*) based on the UUT type/role:
  - a. For Cable Plug UUT: the check fails if the UUT does not send a *Discover Identity* ACK message. [TEST.PD.PROT.ALL.4#12]
  - b. For Sink UUT: the check fails if the UUT does not send a *Sink\_Capabilities* Message. [TEST.PD.PROT.ALL.4#13]
  - c. If the UUT is a Source, the Tester checks the response as follows: [TEST.PD.PROT.ALL.4#14]
    - i. If the VIF field PD\_Port\_Type is set to Provider/Consumer, check that the UUT sends a *Sink\_Capabilities* Message.
    - ii. In PD2 mode, if the VIF field PD\_Port\_Type is set to Provider only, check that the UUT sends a *Reject* Message.
    - iii. In PD3 mode, if the VIF field PD\_Port\_Type is set to Provider only, check that the UUT sends a *Not\_Supported* Message.
13. If the UUT is a Cable Plug, the Tester repeats the test using *Cable Reset* (in lieu of *Hard Reset*).

#### TEST.PD.PROT.ALL.5 Unrecognized Message

**Description:** The Tester checks that the UUT responds correctly to unrecognized message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
Rev2Src		✓	✓		
Rev3ChkdSrc		✓	✓		
Rev2Snk	✓		✓		
Rev3ChkdSnk	✓		✓		
Rev2VcMinEnd1				✓	✓
Rev3VcMinEnd1				✓	✓
Rev2VcMaxEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev2VcMinEnd2				✓	
Rev3VcMinEnd2				✓	
Rev2VcMaxEnd2				✓	
Rev3VcMaxEnd2				✓	

**Test Procedures:**

1. There are 9 possible bring-up procedures:



- a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends the following message, depending on the UUT type/role:
    - a. If the UUT is a Cable Plug, the Tester sends an Extended Message:
      - i. Chunked bit = 1
      - ii. Data Size field set to 26.
      - iii. Message Type field set to 1110b (Reserved, in PD2 mode) or 11111b (Reserved, in PD3 mode).
      - iv. Bytes 0 to 25 of data block contain incrementing values starting at 0x00

- b. If the UUT is a Source or Sink, the Tester sends a Control Message with Message Type field set to 1110b (Reserved, in PD2 mode) or 1111b (Reserved, in PD3 mode).
3. The Tester checks the response message (with an incremented *MessageID*) based on the UUT type/role:
  - a. For Cable Plug UUT, the check fails if the UUT does not ignore the message i.e. message is received within *tChunkReceiverRequest* max.  
[TEST.PD.PROT.ALL.5#1]
  - b. For Source or Sink UUT, the check fails if the UUT does not send *Reject* Message if in PD2 mode and the *Not\_Supported* Message if in PD3 mode.  
[TEST.PD.PROT.ALL.5#2]

#### 5.1.2 PD3 Mode

The tests in this subsection are run only in PD3 mode.

The tests in this subsection are applicable to all UUTs.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (ie, a Provider/Consumer or Consumer/Provider)
- The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### TEST.PD.PROT.ALL3.1 Get\_Status Response

Description: The Tester verifies that the UUT responds correctly to Get\_Status message.

Test Conditions:

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3ChkdSrc</i>		✓	✓		
<i>Rev3UnChkdSrc</i>		✓	✓		
<i>Rev3ChkdSnk</i>	✓		✓		
<i>Rev3UnChkdSnk</i>	✓		✓		
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

Test Procedures:

1. There are 9 possible bring-up procedures:

- a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. If VIF field Product\_Type = Active Cable, the Tester sends a *Discover Identity* Command to the UUT, the checks fails if the UUT doesn't respond with a *Discover Identity* ACK message. [TEST.PD.PROT.ALL3.1#1]
  3. The Tester sends a Get\_Status message to the UUT.
  4. The Tester checks the response message depending on the UUT type/role:
    - a. If the UUT is a Cable Plug [TEST.PD.PROT.ALL3.1#2]
      - i. If the VIF field Product\_Type = Active Cable, the Tester checks that the UUT sends a Status message

- ii. Otherwise, the Tester checks that the UUT sends either Ignore or Status message.
  - b. If the UUT is a Source or Sink, the check fails if the UUT does not respond with either *Not Supported* or Status message. [TEST.PD.PROT.ALL3.1#3]
- 5. The Tester runs additional check for the Status message: [TEST.PD.PROT.ALL3.1#4]
  - a. For the SOP SDB
    - i. Present Input field: Bit 0 = 0 (Reserved)
    - ii. Present Input field: Bits 5...7 (Reserved) are 0s
    - iii. If Bit 3 (Internal Power from Battery) is 0 in Present Input field, then Present Battery Input field shall be 0
    - iv. Event Flags field: Bit 0 and Bits 5...7 (Reserved) are 0s
    - v. Temperature Status field: Bit 0, Bits 3...7 (Reserved) are 0s
    - vi. Power Status field:
      - 1. If DUT's current port power role = Source
        - a. Bit 0, Bits 6...7 (Reserved) are 0s
        - b. Bit1 = 1 if one or more Src\_PDO\_Max\_Current > 3A, and emulated cable is not e-marked
        - c. Bit1 = 0 if emulated cable is e-marked at 5A
        - d. Bit1 = 0 if DUT has captive cable
        - e. Bit1 = 0 if all SRC\_PDO\_Max\_Current <= 3A
      - 2. If DUT's current port power role = Sink, Bits 0..7 are 0s
    - vii. If Bit 2 (OTP event) is 1 in Event Flags field, then Temperature Status field shall be set to 11b (Over Temperature)
    - viii. Power State Change:
      - 1. bits 6..7(reserved) are 0s
      - 2. bits 0..2: value must be less than 7
      - 3. bit 3..5: value must be less than 4.
  - b. For the SOP'/SOP'' SDB
    - i. Flags field: Bits 1...7 = 0 (Reserved)
    - ii. Flags field Bit 0 (not in Thermal Shutdown)
    - iii. VIF field Product\_Type = Active Cable, the number reported in Internal Temp field is less than the Shutdown Temperature reported in Active Cable VDO 2
    - iv. The number reported in Internal Temp field is either 0, or between 1 and 100.

#### *TEST.PD.PROT.ALL3.2 Get\_Manufacturer\_Info Response*

Description: The Tester verifies that the Source UUT responds correctly to Get\_Manufacturer\_Info message.

Test Specific Tester Behavior: N/A

#### Test Conditions:

Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-
---------------	---------------	---------------	------------	-------------------------

					Only eMarker
Rev3ChkdSrc		✓	✓		
Rev3UnChkdSrc		✓	✓		
Rev3ChkdSnk	✓		✓		
Rev3UnChkdSnk	✓		✓		
Rev3VcMinEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev3VcMinEnd2				✓	
Rev3VcMaxEnd2				✓	

#### Test Procedures:

1. There are 9 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF

parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.

- i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a Get\_Manufacturer\_Info message with Manufacturer Info Target set to 0.
3. The Tester checks the response message depending on the UUT type/role:
  - a. If the UUT is a Cable Plug: [TEST.PD.PROT.ALL3.2#1]
    - i. If VIF field Manufacturer\_Info\_Supported = Yes, then the Tester checks that the UUT responds with a Manufacturer\_Info message.
    - ii. Otherwise, the Tester checks that the UUT responds with an Ignore.
  - b. If the UUT is a Source or Sink: [TEST.PD.PROT.ALL3.2#2]
    - i. If VIF field Manufacturer\_Info\_Supported\_Port = Yes, then the Tester checks that the UUT responds with a Manufacturer\_Info message.
    - ii. Otherwise, the Tester checks that the UUT responds with a *Not\_Supported* Message.
4. The Tester runs the following VIF field checks for Manufacturer\_Info message: [TEST.PD.PROT.ALL3.2#3]
  - a. Manufacturer\_Info\_VID(\_SOP)
  - b. Manufacturer\_Info\_PID(\_SOP)

#### TEST.PD.PROT.ALL3.3 Invalid Manufacturer Info Target

**Description:** The Tester sends a *Get\_Manufacturer* Message with an invalid target and verifies correct field values in the response message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
Rev3ChkdSrc		✓	✓		
Rev3UnChkdSrc		✓	✓		
Rev3ChkdSnk	✓		✓		
Rev3UnChkdSnk	✓		✓		
Rev3VcMinEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev3VcMinEnd2				✓	
Rev3VcMaxEnd2				✓	

**Test Procedures:**

1. There are 9 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a *Get\_Manufacturer\_Info* message with Manufacturer Info Target set to 2 and Manufacturer Info Ref set to 0.
3. The Tester checks the response message depending on the UUT type/role:
  - a. If the UUT is a Cable Plug, the check fails if the UUT does not respond with either *Ignore* or *Manufacturer\_Info* Message [TEST.PD.PROT.ALL3.3#1]



- b. If the UUT is a Source or Sink, the check fails if the UUT does not respond with either *Not\_Supported* or *Manufacturer\_Info* Message.  
[TEST.PD.PROT.ALL3.3#2]
4. The Tester verifies Manufacturer\_Info message and performs below checks.
  - a. Manufacturer String in Manufacturer Info Data Block is null terminated ASCII text string "Not Supported".
  - b. VID field is 0xFFFF and PID field is 0x0000 .  
[TEST.PD.PROT.ALL3.3#3]

#### TEST.PD.PROT.ALL3.4 Invalid Manufacturer Info Ref

**Description:** The Tester sends a *Get\_Manufacturer* Message with an invalid reference and verifies correct field values in the response message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
Rev3ChkdSrc		✓	✓		
Rev3UnChkdSrc		✓	✓		
Rev3ChkdSnk	✓		✓		
Rev3UnChkdSnk	✓		✓		
Rev3VcMinEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev3VcMinEnd2				✓	
Rev3VcMaxEnd2				✓	

**Test Procedures:**

1. There are 9 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink

- COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
- e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a *Get\_Manufacturer\_Info* Message with Manufacturer Info Target set to 1 and Manufacturer Info Ref set to 8.
  3. The Tester checks the response message depending on the UUT type/role:
    - a. If the UUT is a Cable Plug, the check fails if the UUT does not respond with either Ignore or *Manufacturer\_Info* Message. [TEST.PD.PROT.ALL3.4#1]
    - b. If the UUT is a Source or Sink, the check fails if the UUT does not respond with either *Not\_Supported* or *Manufacturer\_Info* Message. [TEST.PD.PROT.ALL3.4#2]
  4. The Tester verifies Manufacturer\_Info message and performs below checks.
    - a. Manufacturer String in Manufacturer Info Data Block is null terminated ASCII text string "Not Supported".
    - b. VID field is 0xFFFF and PID field is set to 0x000. [TEST.PD.PROT.ALL3.4#3]

#### TEST.PD.PROT.ALL3.5 Chunked Extended Message Response

**Description:** The Tester verifies that the UUT receives a chunked extended message correctly by sending messages to request chunks.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-
---------------	---------------	---------------	------------	-------------------------

					Only eMarker
Rev3ChkdSrc		✓	✓		
Rev3ChkdSnk	✓		✓		
Rev3VcMinEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev3VcMinEnd2				✓	
Rev3VcMaxEnd2				✓	

#### Test Procedures:

1. There are 9 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.

- i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends the first chunk of a chunked extended message to the UUT.
  - a. Chunked bit = 1
  - b. Data Size field set to 260.
  - c. Message Type set to 11111b
  - d. Bytes 0 to 259 of data block contain incrementing values (mod 256) starting at 0x00
3. The tester checks the response
  - a. If the UUT is a Cable Plug, the Tester check the response: [TEST.PD.PROT.ALL3.5#1]
    - i. The check fails if the UUT does not respond with either Ignore or a request for a Chunk of Data Block message.
    - ii. If the message is ignored, the test ends here.  
The check fails if VIF field Chunking\_Implemented = YES
    - iii. If a request for a Chunk of a Data Block is received
      1. The check fails if the request for a Chunk of a Data Block is not received within *tChunkReceiverRequest* max.
      2. The check fails if VIF field Chunking\_Implemented = NO
  - b. If the UUT is a Source or Sink, the Tester check the response: [TEST.PD.PROT.ALL3.5#2]
    - i. The check fails if Sink or Source UUT does not respond with either *Not\_Supported* or a request for a Chunk of Data Block message.
    - ii. If a *Not\_Supported* Message is received, the test ends here.
      1. The check fails if VIF field Chunking\_Implemented\_SOP = YES
      2. The check fails if *Not\_Supported* Message is not received between *tChunkingNotSupported* min and *tChunkingNotSupported* max.
    - iii. If a request for a Chunk of the Data Block is received
      1. The check fails if a request for a Chunk of a Data Block is not received within *tChunkReceiverRequest* max
      2. The check fails if VIF field Chunking\_Implemented\_SOP = NO
  - c. The check fails if this is not the first chunk and the UUT does not respond with a request for chunk message. If the UUT does not respond with a request for chunk message, the test stops here. Else, upon receipt of a request response from the UUT, the Tester sends the requested chunk to the UUT. This step is repeated until the Tester has sent all 10 chunk messages. The Tester runs the following additional checks on the request chunk message: [TEST.PD.PROT.ALL3.5#3]
    - i. For Message Header
      1. Bit 15 Extended bit = 1
      2. Bits 14...12 Number of Data Objects = 1
      3. Bits 4...0 Message Type = 11111b
    - ii. For Extended Message Header
      1. Bits 14...11 Chunk Number = next chunk in the series
      2. Bit 10 Request Chunk = 1

3. Bits 8...0 Data Size = 0
4. The Tester check fails if:
  - a. The UUT is a Source or Sink, and it does not send a *Not\_Supported* Message within *tReceiverResponse* after it has requested the 10 chunk messages.
  - b. The UUT is a Cable Marker, and it does not ignore the completed Extended message.

[TEST.PD.PROT.ALL3.5#4]

#### TEST.PD.PROT.ALL3.6 ChunkSenderResponseTimer Timeout

**Description:** The Tester verifies that the UUT recovers correctly after the Tester stops sending chunked messages in the middle.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
Rev3ChkdSrc		✓	✓		
Rev3ChkdSnk	✓		✓		
Rev3VcMinEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev3VcMinEnd2				✓	
Rev3VcMaxEnd2				✓	

#### Test Procedures:

1. There are 9 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section need to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section need to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).

- e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends the first chunk of a chunked extended message to the UUT.
    - a. Chunked bit = 1
    - b. Data Size field set to 260.
    - c. Message Type set to 11111b.
    - d. Bytes 0 to 259 of data block contain incrementing values (mod 256) starting at 0x00.
  3. The tester checks the response.
    - a. If the UUT is a Cable Plug, the check fails if the UUT does not respond with either **Ignore** or a request for a Chunk of Data Block message.  
[TEST.PD.PROT.ALL3.6#1]
    - b. If the UUT is a Source or Sink, the Tester checks response:  
[TEST.PD.PROT.ALL3.6#2]
      - i. The check fails if Sink or Source UUT does not respond with either **Not\_Supported** or a request for a Chunk of Data Block message.
      - ii. The check fails if a **Not\_Supported** Message received, and it is not received between *tChunkingNotSupported* min and *tChunkingNotSupported* max.
      - iii. The check fails if a request for a Chunk of a Data Block is received, and it is not received within *tChunkReceiverRequest* max.
    - c. The test stops here if the UUT does not respond with a request for chunk message. Upon receipt of a request response from the UUT, the Tester sends the requested chunk to the UUT. This step is repeated until the Tester has sent 4 chunk messages and it intentionally does not send the 5<sup>th</sup> chunk message to the UUT.
    - d. The Tester runs the following additional checks the request chunk message:  
[TEST.PD.PROT.ALL3.6#3]
      - i. For Message Header

1. Number of Data Objects = 1
2. Message Type = 1111b
- ii. For Extended Message Header
  1. Bits 14...11 Chunk Number = next chunk in the series
  2. Bit 10 Request Chunk = 1
  3. Bits 8...0 Data Size = 0
4. The Tester waits for *tChunkSenderResponse* max + *tCtsChunkMargin* before re-sending to the UUT the first chunk message of the same chunked extended message defined in step 2.
5. The tester checks the response.
  - a. If the UUT is a Cable Plug, the check fails if the UUT does not respond with either **Ignore** or a request for a Chunk of Data Block message.  
[TEST.PD.PROT.ALL3.6#4]
  - b. If the UUT is a Source or Sink, the Tester checks the response  
[TEST.PD.PROT.ALL3.6#5]
    - i. The check fails if Sink or Source UUT does not respond with either *Not\_Supported* or a request for a Chunk of Data Block message.
    - ii. The check fails if a *Not\_Supported* Message received, it is not received between *tChunkingNotSupported* min and *tChunkingNotSupported* max.
    - iii. The check fails if a request for a Chunk of a Data Block is received, it is not received within *tChunkReceiverRequest* max.
  - c. Upon receipt of the response from the UUT, the Tester sends the requested chunk to the UUT.
  - d. The Tester runs the following additional checks on the request chunk message:  
[TEST.PD.PROT.ALL3.6#6]
    - i. For Message Header
      1. Number of Data Objects = 1
      2. Message Type = 1111b
    - ii. For Extended Message Header
      1. Bits 14...11 Chunk Number = next chunk in the series
      2. Bit 10 Request Chunk = 1
      3. Bits 8...0 Data Size = 0

#### TEST.PD.PROT.ALL3.7 Security Messages Supported

**Description:** The Tester verifies the support of security messages is as specified in the VIF.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
Rev3ChkdSrc		✓	✓		
Rev3UnChkdSrc		✓	✓		
Rev3ChkdSnk	✓		✓		
Rev3UnChkdSnk	✓		✓		



<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

#### Test Procedures:

1. There are 9 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - i. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.

2. The Tester sends a *Security\_Request* Message, in which SRQDB is set to {0x01, 0x81, 0x00, 0x00}, to the UUT.
3. The Tester checks the response:
  - a. If VIF field *Security\_Msgs\_Supported(\_SOP)* is set to NO, the Tester checks that the UUT replies *Not\_Supported* (if VIF field *VIF\_Product\_Type* not set to Cable) or Ignored (if VIF field *VIF\_Product\_Type* set to Cable). The test stops here. **[TEST.PD.PROT.ALL3.7#1]**
  - b. If VIF field *Security\_Msgs\_Supported(\_SOP)* is set to YES, the Tester checks a *Security\_Response* Message is received within *tDigestSent* (different values for **Unchunked** and **Chunked**). **[TEST.PD.PROT.ALL3.7#2]**. In this case, the UUT shall also run Authentication Compliance tests in CV.

#### *TEST.PD.PROT.ALL3.8 Get Revision Response*

**Description:** The tester verifies that the UUT responds correctly to *Get\_Revision* Message

**Test Specific Tester Behavior:** N/A

#### **Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3ChkdSrc</i>		✓	✓		
<i>Rev3ChkdSnk</i>	✓		✓		
<i>Rev3VcMidEnd1</i>				✓	✓
<i>Rev3VcMidEnd2</i>				✓	

#### **Test Procedures:**

1. There are 9 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field *PD\_Port\_Type* set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field *PD\_Port\_Type* set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it

- runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
- e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  - f. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - g. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - h. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - i. The UUT has VIF parameter PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends the *Get\_Revision* Messages requesting the *Revision* Message from the UUT
  3. The Tester checks: [TEST.PD.PROT...#1]
    - a. If product is Passive Cable (*Table 2 Cable Types*), the UUT may respond with either *Revision* Message or Ignored. If any other responses received, the Tester fails the check, and the test stops here. If the UUT Ignores the message, the Tester passes the check, and the test stops here. If the *Revision* Message received, the Tester goes to the next step.
    - b. If product is VPD (*Table 1 UUT Types*), the UUT may respond with either *Revision* Message or Ignored. If any other responses received, the Tester fails the check, and the test stops here. If the UUT Ignores the message, the Tester passes the check, and the test stops here. If the *Revision* Message received, the Tester goes to the next step.
    - c. Else (all other products like active cable, producer, consumer and etc.) the UUT shall reply with the *Revision* message. If the Tester does not receive the *Revision* Message, the Tester fails the check, and the test stops here.
  4. The Tester checks the *Revision* Message details: [TEST.PD.PROT...#2]
    - a. The Number of Data objects in *Revision* Message header is 1
    - b. In RMD0, B31...28 Revision.major matches to the VIF PD\_Spec\_Revision\_Major field
    - c. In RMD0, B27...24 Revision.minor matches to the VIF PD\_Spec\_Revision\_Minor field

- d. In RMDO, B23...20 Version.major matches to the VIF PD\_Spec\_Version\_Major field
- e. In RMDO, B19...16 Version.minor matches to the VIF PD\_Spec\_Version\_Minor field

In RMDO, B15...0 are set to Zero

## 5.2 Test Procedures for Source and Sink Capable UUTs

### 5.2.1 PD2 and PD3 Modes

None

### 5.2.2 PD3 Mode

The tests in this subsection are run only in PD3 mode.

The tests in this subsection are applicable to UUT with VIF field VIF\_Product\_Type set to Port.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (i.e., a Provider/Consumer or Consumer/Provider)
- The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### TEST.PD.PROT.PORT3.1 Get\_Battery\_Status Response

Description: The Tester verifies that the Source UUT responds correctly to *Get\_Battery\_Status* Message

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓
Rev3ChkdSnk	✓		✓

Test Procedures:

1. There are 5 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).

- c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
2. If the Tester behaves as a Sink, it sends a *Get\_Source\_Cap\_Extended* Message to the UUT. Else go to step 4.
  3. The check fails: [TEST.PD.PROT.PORT3.1#1]
    - a. If VIF field PD\_Port\_Type is not Consumer Only, and the UUT does not send either *Not\_Supported* or *Source\_Capability\_Extended* message.
    - b. If VIF field PD\_Port\_Type is Consumer Only, and the UUT does not send *Not\_Supported*.
  4. If the Tester behaves as a Source, it sends a *Get\_Sink\_Cap\_Extended* Message to the UUT.
  5. The check fails: [TEST.PD.PROT.PORT3.1#1]
    - a. If VIF field PD\_Port\_Type is not Provider Only, and the UUT does not send the *Sink\_Capability\_Extended* message.
    - b. If VIF field PD\_Port\_Type is Provider Only, and the UUT does not send *Not\_Supported*.
  6. The Tester sends a *Get\_Battery\_Status* Message to the UUT, with Battery Status Ref field set to 0.
  7. The check fails if the UUT does not send either *Not\_Supported* or *Battery\_Status* Message. [TEST.PD.PROT.PORT3.1#2]
  8. If the UUT has sent *Not\_Supported* as a response to *Get\_Battery\_Status*, the Tester checks the response: [TEST.PD.PROT.PORT3.1#3]
    - a. The check fails if the UUT has sent *Source\_Capabilities\_Extended* or *Sink\_Capabilities\_Extended* Message as a response to *Get\_Source\_Cap\_Extended* or *Get\_Sink\_Cap\_Extended* and the Number of Batteries/Battery Slots field in the message has non-zero value.
    - b. The check fails if the values of VIF fields Num\_Fixed\_Batteries and Num\_Swappable\_Battery\_Slots are non-zero.
  9. If the UUT has sent *Battery\_Status* in response to *Get\_Battery\_Status*, the Tester runs the following additional field checks for *Battery\_Status* Message: [TEST.PD.PROT.PORT3.1#4]
    - a. Bits 7...0 of the BSDO are 0
    - b. Battery Info field:

- i. Bits 7...4 (Reserved) are 0
  - ii. Bit 0 (Invalid Battery Reference field):
    1. If *Source\_Capabilities\_Extended* or *Sink\_Capabilities\_Extended* is received as a response to *Get\_Source\_Cap\_Extended* or *Get\_Sink\_Cap\_Extended*, check against Number of Batteries/Battery Slots field in the message.
    2. Check against VIF fields Num\_Fixed\_Batteries and Num\_Swappable\_Battery\_Slots
  - iii. If Bit 0 (Invalid Battery Reference bit) is 0b and Battery Status Ref field (in the *Get\_Battery\_Status* Message) has a value between 0 and 3, then check that the Bit 1 (Battery is Present bit) is set to 1b.
  - iv. If Bit 0 (Invalid Battery Reference bit) is 1b, then check that the Bit 1 (Battery is Present bit) is set to 0b.
  - v. If Bit 1 (Battery is Present bit) is set to 1b, then check that Bits 3...2 (Battery Charging Status field) is not 11b.
  - vi. If Bit 1 (Battery is Present bit) is set to 0b, then check that Bits 3...2 (Battery Charging Status field) is 00b.
10. Rerun steps 8 and 9 with Battery Status Ref (in the *Get\_Battery\_Status* Message) set to 1

– 7

#### *TEST.PD.PROT.PORT3.2 Invalid Battery Status Reference*

**Description:** The Tester sends a *Get\_Battery\_Status* Message with an invalid battery reference to the UUT and verifies correct field values in the response message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓
Rev3ChkdSnk	✓		✓

**Test Procedures:**

1. There are 5 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a



Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).

- e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
2. If the Tester behaves as a Sink, it sends a *Get\_Source\_Cap\_Extended* Message to the UUT. Else go to step 4.
3. The check fails: [TEST.PD.PROT.PORT3.2#1]
  - a. If VIF field PD\_Port\_Type is not Consumer Only, and the UUT does not send either the *Not\_Supported* or *Source\_Capabilities\_Extended* Message.
  - b. If VIF field PD\_Port\_Type is Consumer only, and the UUT does not send *Not\_Supported*.
4. If the Tester behaves as a Source, it sends a *Get\_Sink\_Cap\_Extended* Message to the UUT.
5. The check fails: [TEST.PD.PROT.PORT3.2#1]
  - a. If VIF field PD\_Port\_Type is not Provider Only, and the UUT does not send *Sink\_Capabilities\_Extended* Message.
  - b. If VIF field PD\_Port\_Type is Provider Only, and the UUT does not send *Not\_Supported*.
6. The Tester sends a *Get\_Battery\_Status* Message with Battery Status Ref field set to 8.
7. The check fails if the UUT does not send either the *Not\_Supported* or *Battery\_Status* Message. [TEST.PD.PROT.PORT3.2#2]
8. If the UUT has sent *Not\_Supported* as a response to *Get\_Battery\_Status*, the Tester check the response: [TEST.PD.PROT.PORT3.2#3]
  - a. The check fails if the UUT has sent *Source\_Capabilities\_Extended* or *Sink\_Capabilities\_Extended* Message as a response to *Get\_Source\_Cap\_Extended* or *Get\_Sink\_Cap\_Extended* Message and the Number of Batteries/Battery Slots field in the message has non-zero value
  - b. The check fails if the values of VIF fields Num\_Fixed\_Batteries and Num\_Swappable\_Battery\_Slots are non-zero.
9. If the UUT has sent *Battery\_Status* in response to *Get\_Battery\_Status*, the Tester runs the following additional field checks for *Battery\_Status* Message: [TEST.PD.PROT.PORT3.2#4]
  - a. Bits 7..0 of the BSDO are 0
  - b. Battery Info field:
    - i. Check that Bit 0 (Invalid Battery Reference field) is 1b
    - ii. Check that Bits 7..1 are 0

#### *TEST.PD.PROT.PORT3.3 Get\_Battery\_Cap Response*

Description: The Tester verifies that the Source UUT responds correctly to *Get\_Battery\_Cap* Message



Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓
Rev3UnchkdSrc		✓	✓
Rev3ChkdSnk	✓		✓
Rev3UnchkdSnk	✓		✓

Test Procedures:

- There are 5 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
- If the Tester behaves as a Sink, it sends a *Get\_Source\_Cap\_Extended* Message to the UUT. Else go to step 4.
- The check fails. [TEST.PD.PROT.PORT3.3#1]
  - If VIF field PD\_Port\_Type is not Consumer only, and the UUT does not send the *Not\_Supported* or Source\_Capabilities\_Extended Message.
  - If VIF field PD\_Port\_Type is Consumer only, and the UUT does not send *Not\_Supported*
- If the Tester behaves as a Source, it sends a *Get\_Sink\_Cap\_Extended* Message to the UUT.
- The check fails: [TEST.PD.PROT.PORT3.3#1]

- a. If VIF field PD\_Port\_Type is not Provider Only, and the UUT does not send *Sink\_Capabilities\_Extended* Message
  - b. If VIF field PD\_Port\_Type is Provider Only, and the UUT does not send *Not\_Supported*
6. The Tester sends a *Get\_Battery\_Cap* Message to the UUT, with Battery Cap Ref field set to 0.
7. The Tester checks the response [TEST.PD.PROT.PORT3.3#2]
  - a. If either Num\_Fixed\_Batteries is not set to 0 or Num\_Swappable\_Battery\_Slots is not set to 0, the Tester checks that the UUT sends *Battery\_Capabilities* Message
  - b. Otherwise, the Tester checks that the UUT sends either:
    - i. *Not\_Supported*, or
    - ii. A *Battery\_Capabilities* Message with:
      1. VID field set to FFFF and PID field set to 0x0000.
      2. The Invalid Reference bit set to 1.
      3. The reserved bits set to 0.
8. The Tester runs the following additional field checks for *Battery\_Capabilities* Message: [TEST.PD.PROT.PORT3.3#3]
  - a. Battery Type field:
    - i. Check that Bit 0 (Invalid Battery Reference field) matches:
      1. If the *Source\_Capabilities\_Extended* or *Sink\_Capabilities\_Extended* Message is received as a response to the *Get\_Source\_Cap\_Extended* or *Get\_Sink\_Cap\_Extended* Message, check against Number of Batteries/Battery Slots field in the message.
      2. If the *Not\_Supported* Message is received as a response to *Get\_Battery\_Cap* Message, check against VIF fields Num\_Fixed\_Batteries and Num\_Swappable\_Battery\_Slots
    - ii. Check that Bits 1...7 are 0
9. Rerun steps 7 and 8 with Battery Cap Ref (in the *Get\_Battery\_Cap* Message) set to 1 – 7

#### TEST.PD.PROT.PORT3.4 Invalid Battery Capabilities Reference

**Description:** The Tester sends a *Get\_Battery\_Capabilities* Message with an invalid battery reference to the UUT and verifies correct field values in the response message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓
Rev3UnchkdSrc		✓	✓
Rev3ChkdSnk	✓		✓
Rev3UnchkdSnk	✓		✓

**Test Procedures:**

1. There are 5 possible bring-up procedures:

- a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
2. If the Tester behaves as a Sink, it sends a *Get\_Source\_Cap\_Extended* Message to the UUT. Else go to step 4.
  3. The check fails: [TEST.PD.PROT.PORT3.4#1]
    - a. If VIF field PD\_Port\_Type is not Consumer Only, and the UUT does not send either the *Not\_Supported* or *Source\_Capabilities\_Extended* Message.
    - b. If VIF field PD\_Port\_Type is Consumer only, and the UUT does not send *Not\_Supported*.
  4. If the Tester behaves as a Source, it sends a *Get\_Sink\_Cap\_Extended* Message to the UUT.
  5. The check fails: [TEST.PD.PROT.PORT3.4#1]
    - a. If VIF field PD\_Port\_Type is not Provider Only, and the UUT does not send the *Sink\_Capabilities\_Extended* Message.
    - b. If VIF field PD\_Port\_Type is Provider Only, and the UUT does not send *Not\_Supported*.
  6. The Tester sends a *Get\_Battery\_Cap* Message to the UUT, with Battery Cap Ref field set to 8.
  7. The check fails if the UUT does not send either the *Not\_Supported* or *Battery\_Capabilities* Message. [TEST.PD.PROT.PORT3.4#2]
  8. The Tester runs the following additional field checks for the *Battery\_Capabilities* Message: [TEST.PD.PROT.PORT3.4#3]
    - i. VID field set to FFFF and PID field set to 0x0000.
    - ii. The Invalid Reference bit set to 1.

- iii. The reserved bits set to 0.

#### *TEST.PD.PROT.PORT3.5 Get\_Country\_Codes Response*

**Description:** The Tester verifies that the Source UUT responds correctly to the *Get\_Country\_Codes* Message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓
Rev3UnchkdSrc		✓	✓
Rev3ChkdSnk	✓		✓
Rev3UnchkdSnk	✓		✓

**Test Procedures:**

1. There are 5 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
2. The Tester sends a *Get\_Country\_Codes* Message to the UUT.
3. The check fails if the UUT does not send either the *Not\_Supported* or the *Country\_Codes* Message. [TEST.PD.PROT.PORT3.5#1]
4. The Tester runs the following additional field checks for CCDB: [TEST.PD.PROT.PORT3.5#2]

- a.  $\text{Length} = (\text{Data Size}) / 2 - 1$
- b. All other bytes are ascii codes of capital letters (65 - 90 inclusive)

#### *TEST.PD.PROT.PORT3.6 Get\_Country\_Info Response*

**Description:** The Tester verifies that the Source UUT responds correctly to Get\_Country\_Info message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓
Rev3UnchkdSrc		✓	✓
Rev3ChkdSnk	✓		✓
Rev3UnchkdSnk	✓		✓

**Test Procedures:**

1. There are 5 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
2. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester

behaves as a Sink only and it runs bring-up procedure with the Tester sends a *Get\_Country\_Codes* Message to the UUT.

3. The check fails if the UUT does not send either *Not\_Supported* or *Country\_Codes* Message. [TEST.PD.PROT.PORT3.6#1]
4. If a *Not\_Supported* Message is received, the test stops here. Otherwise, the Tester sends a *Get\_Country\_Info* Message to the UUT.
5. The check fails if the UUT does not send the *Country\_Info* Message. [TEST.PD.PROT.PORT3.6#2]
6. The Tester runs the following additional field checks for CIDB: [TEST.PD.PROT.PORT3.6#3]
  - a. Country Code field (Byte 0 and Byte 1) is the same as that in the *Get\_Country\_Info* message sent by the Tester.
  - b. Byte 2 and Byte 3 (Reserved) are 0.
7. Rerun the test with the bring-up procedures in Step-1 with the Unchunked Extended Messages Supported field set to 1b in the Tester sent Request (RDO) or *Source\_Capabilities* PDO.

#### TEST.PD.PROT.PORT3.7 Unchunked\_Extended\_Messages\_Supported

**Description:** The Tester verifies that that if Unchunked\_Extended\_Messages\_Supported is set to yes, the UUT can receive Extended Messages with Data Size > MaxExtendedMsgLegacyLen bytes in a single, Unchunked Message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

This test is not applicable if Unchunked\_Extended\_Messages\_Supported is set to no.

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3UnChkdSrc		✓	✓
Rev3UnChkdSnk	✓		✓

**Test Procedures:**

1. There are 5 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only. The Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (Dual-Role Power bit B29 in the single PDO is set to 1b).
  - c. The UUT has VIF field PD\_Port\_Type set to Provider Only. The Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, each test under this section needs to be run twice, first time the Tester behaves as a Consumer/Provider and it runs bring-up procedure with the UUT as a Sink



- COMMON.PROC.BU.8; second time the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 1b).
- e. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (Dual-Role Power bit B29 in the single PDO is set to 0b).
  2. The UUT has VIF field PD\_Port\_Type set to DRP, each test under this section needs to be run twice, first time the Tester behaves as a Source only and it runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2; second time the Tester behaves as a Sink only and it runs bring-up procedure with the Tester sends an extended message to the UUT, with Data Size set to 260 and Message Type set to 1111b. Bytes 0 to 259 of data block contain incrementing values (mod 256) starting at 0x00.
  3. If a *GoodCRC* is not received, the check fails. [TEST.PD.PROT.PORT3.7#1]
  4. If a *Not\_Supported* Message is not received within *tReceiverResponse* max, the check fails. This delay is measured from the time the last bit of the extended message EOP has been transmitted to the time the first bit of the *Not\_Supported* Message preamble has been received. [TEST.PD.PROT.PORT3.7#2]

### 5.3 Test Procedures for Source Capable UUT

#### 5.3.1 PD2 and PD3 Modes

The tests in this subsection are run firstly in PD2 mode and secondly in PD3 mode.

The tests in this subsection are applicable to UUT with VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer and DRP.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (i.e., a Provider/Consumer or Consumer/Provider) The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- In the PD2 mode:
  - the Tester rejects *VCONN\_Swap*.
 In the PD3 mode:
  - if the Tester is a V<sub>CONN</sub> Source, the Tester accepts the VCONN\_Swap Message.
  - else, the Tester rejects the *VCONN\_Swap* Message.
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### TEST.PD.PROT.SRC.1 *Get\_Source\_Cap Response*

Description: The Tester verifies that the Source UUT responds correctly to *Get\_Source\_Cap* Message.

Test Specific Tester Behavior: N/A

Test Conditions:



	Consumer Only	Provider Only, C/P, DRP, P/C
Rev2Src		✓
Rev3ChkdSrc		✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester runs checks for the VIF parameter Has\_Invariant\_PDOs and *Source Capabilities*:
  - a. If VIF parameter Has\_Invariant\_PDOs is set to N and the current *Source Capabilities* do not match those in the VIF, run common procedure COMMON.PROC.PD.18, then the test proceeds to the next step.
  - b. Otherwise, the test proceeds to the next step.
3. The Tester sends a *Get\_Source\_Cap* Message to the UUT.
4. The check fails if the Tester does not receive *Source\_Capabilities* Message.  
[TEST.PD.PROT.SRC.1#1]
5. Once the Tester receives *Source\_Capabilities* Message, the Tester establishes a PD contract using common procedure COMMON.PROC.PD.11.

#### *TEST.PD.PROT.SRC.2 Get\_Source\_Cap No Request*

Description: The Tester sends a *Get\_Source\_Cap* Message to the UUT. After receiving a *Source\_Capabilities* Message, the Tester intentionally does not send the *Request* Message to force a *SenderResponse* Timer timeout on the Source UUT. The Tester verifies correct implementation of this timer.

Test Specific Tester Behavior: N/A

#### Test Conditions:

	Consumer Only	Provider Only, C/P, P/C, DRP
Rev2Src		✓
Rev3ChkdSrc		✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester sends a *Get\_Source\_Cap* Message to the UUT.

3. The check fails if the Tester does not receive *Source\_Capabilities* Message.  
[TEST.PD.PROT.SRC.2#1]
4. The Tester does not send a *Request* Message after receiving *Source\_Capabilities* Message.
5. The Tester checks that a *Hard Reset* is completed between *tSenderResponse* min and *tSenderResponse* max + *tHardResetComplete* max (see Table 19 Timing Table & Calculations) the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the last bit of *Hard Reset* EOP has been received. [TEST.PD.PROT.SRC.2#2]

#### TEST.PD.PROT.SRC.3 *SenderResponseTimer Deadline*

**Description:** As a Sink, the Tester checks that the UUT accepts a *Request* Message sent at the deadline limit of *tSenderResponse* min in reply to a *Source\_Capabilities* Message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, C/P	Provider Only, DRP, P/C
Rev2Src		✓
Rev3ChkdSrc		✓

#### **Test Procedures:**

1. The test starts in a disconnected state.
2. The Tester applies Rd and it waits for V<sub>BUS</sub> for *tNoResponse* max.
3. The check fails if the first bit of a *Source\_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after V<sub>BUS</sub> present. [TEST.PD.PROT.SRC.3#1]
4. The Tester responds with a *Request* Message at *tCtsSnkRequest* (see Table 19 Timing Table & Calculations) after the *GoodCRC* Message.
5. The Tester checks the response [TEST.PD.PROT.SRC.3#2]
  - a. The check fails if the UUT does not send either *Accept*, *Reject* or *Wait* Message
  - b. The check fails if the UUT sent a *Hard Reset*

#### TEST.PD.PROT.SRC.4 *Reject Request*

**Description:** As a Sink, the Tester requests 200% of the offered current or power under the terms of each PDO on the UUT, and verifies the request is rejected by the UUT.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, C/P	Provider Only, DRP, P/C
Rev2Src		✓
Rev3ChkdSrc		✓

#### **Test Procedures:**

1. The test starts in a disconnected state.
2. The Tester applies Rd and it waits for V<sub>BUS</sub> for *tNoResponse* max.

3. The check fails if the first bit of a *Source\_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after  $V_{BUS}$  present. [TEST.PD.PROT.SRC.4#1]
4. The Tester runs checks for the VIF parameter Has\_Invariant\_PDOs and *Source Capabilities*:
  - a. If VIF parameter Has\_Invariant\_PDOs is set to N and the current *Source Capabilities* do not match those in the VIF, run common procedure COMMON.PROC.PD.18, then the test proceeds to the next step.
  - b. Otherwise, the test proceeds to the next step.
5. The Tester sends a Request (starting from the first PDO), requesting 200% of the current or power offered.
6. The check fails if a *Reject* Message is not received. [TEST.PD.PROT.SRC.4#2]
7. Restart the test from Step 1 by sending Request to the next PDO.

#### TEST.PD.PROT.SRC.5 Reject Request Invalid Object Position

**Description:** As a Sink, the Tester sends a Request with an invalid object position, and verifies the request is rejected by the UUT.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, C/P	Provider Only, DRP, P/C
Rev2Src		✓
Rev3ChkdSrc		✓

**Test Procedures:**

1. The test starts in a disconnected state.
2. The Tester applies  $R_d$  and it waits for  $V_{BUS}$  present.
3. The check fails if the first bit of a *Source\_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after  $V_{BUS}$  present. [TEST.PD.PROT.SRC.5#1]
4. The Tester sends a *Request* Message using the VIF field Num\_Src\_PDOs:
  - a. If Num\_Src\_PDOs = 7, then set Object Position field to 0
  - b. Otherwise, set Object Position field to 1 + Num\_Src\_PDOs
5. The check fails if the UUT does not respond with a *Reject* Message. [TEST.PD.PROT.SRC.5#2]

#### TEST.PD.PROT.SRC.6 Atomic Message Sequence – Request

**Description:** As a Sink, the Tester sends a *Get Sink Cap* Message instead of a *Request* Message and verifies that the UUT correctly sends a *Soft Reset* to recover from this error.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, C/P	Provider Only, DRP, P/C
Rev2Src		✓
Rev3ChkdSrc		✓

#### Test Procedures:

1. The test starts in a disconnected state.
2. The Tester applies Rd and it waits for V<sub>BUS</sub> present.
3. The check fails if the first bit of a *Source\_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after V<sub>BUS</sub> present. [TEST.PD.PROT.SRC.6#1]
4. The Tester sends a *Get\_Sink\_Cap* Message instead of a *Request* Message.
5. The check fails if a *Soft\_Reset* Message is not received within *tProtErrSoftReset*. [TEST.PD.PROT.SRC.6#2]
6. The Tester sends an *Accept* Message to the UUT (as a response to *Soft\_Reset* Message).
7. The check fails if a *Source\_Capabilities* Message with a *MessageID* = 1 is not received within *tTypeCSinkWaitCap* max from the last bit of the EOP of the *GoodCRC* Message. [TEST.PD.PROT.SRC.6#3]

#### *TEST.PD.PROT.SRC.7 DR\_Swap*

Description: As a Sink, the Tester sends a *DR\_Swap* Message, and verifies that the UUT responds correctly.

#### Test Specific Tester Behavior:

- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message

#### Test Conditions:

	Consumer Only	Provider Only, C/P, P/C, DRP
Rev2Src		✓
Rev3ChkdSrc		✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester instead sends *Request* Message with B25 (USB Communication Capable) set to 1b.
3. The Tester sends a *DR\_Swap* Message.
4. The Tester checks the response. [TEST.PD.PROT.SRC.7#1]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *DR\_Swap* every *tDRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *DR\_Swap* for *nCtsWaitCount* times.
  - b. The check fails if:
    - i. In PD2, the UUT responds with a *Reject* Message and the VIF field DR\_Swap\_To\_UFP\_Supported = Yes.
    - ii. In PD3, the UUT responds with either:
      1. *Not\_Supported* and (VIF field DR\_Swap\_To\_UFP\_Supported = Yes or VIF field DR\_Swap\_To\_DFP\_Supported = Yes) or

2. *Reject* and the VIF fields DR\_Swap\_To\_UFP\_Supported = Yes  
If the check fails, the test ends here.
- c. The check fails if the UUT responds with an *Accept* Message and the VIF field DR\_Swap\_To\_UFP\_Supported = No.
5. If VIF DR\_SWAP\_To\_UFP\_Supported=N, test ends here. Otherwise, the Tester requests a further *DR\_Swap*.
6. The Tester checks the response. [TEST.PD.PROT.SRC.7#2]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *DR\_Swap* every  $tDRSwapWait + tCtsWait$  for  $nCtsWaitCount$  times. The check fails and the test ends here if the Tester has resent *DR\_Swap* for  $nCtsWaitCount$  times.
  - b. The check fails if :
    - i. In PD2, the UUT responds with a *Reject* Message and the VIF field DR\_Swap\_To\_DFP\_Supported = Yes.
    - ii. In PD3, the UUT responds with either.
      1. *Not\_Supported* and (the VIF fields DR\_Swap\_To\_DFP\_Supported = Yes or the VIF fields DR\_Swap\_To\_UFP\_Supported = Yes) or
      2. *Reject* and the VIF fields DR\_Swap\_To\_DFP\_Supported = Yes
  - c. The check fails if the UUT responds with an *Accept* Message and the VIF field DR\_Swap\_To\_DFP\_Supported = No.

#### TEST.PD.PROT.SRC.8 VCONN\_Swap Response

**Description:** As a Sink, the Tester sends a *VCONN\_Swap* message, and verifies that the UUT responds correctly.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
Rev2Src		✓
Rev3ChkdSrc		✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7. If the UUT does not accept VCONN\_Swap to become VCONN Source, the test passes and stops here.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.  
The Tester keeps monitoring the VCONN voltage throughout this test.
2. If VIF field Captive\_Cable = No, the Tester presents Ra to the non-CC line (of the UUT). If a SOP' *Discover Identity* Command is received from the UUT, the Tester replies to as Passive Cable with VBUS Current Handling Capability set to 5A and Maximum VBUS Voltage set to 50V.
3. If UUT sends the *VCONN\_Swap* Message, the Tester rejects the *VCONN\_Swap* Message and the Tester sends a *VCONN\_Swap* Message.

4. The Tester checks the response. [TEST.PD.PROT.SRC.8#1]
  - a. If the UUT responds with a *Wait* Message:
    - i. The check fails if the UUT responds with a *Wait* Message in PD3 mode and the Test stops here.
    - ii. If the UUT responds with a *Wait* Message in PD2 mode, the Tester resends *VCONN\_Swap* every *tVCONNSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *VCONN\_Swap* for *nCtsWaitCount* times in PD2 mode.
  - b. If the UUT responds with an *Accept* Message, the Tester checks that *VCONN\_Swap\_To\_Off\_Supported* = YES
  - c. If the UUT responds with a *Reject* Message:
    - i. The check fails in PD3 mode.
    - ii. The Tester checks *VCONN\_Swap\_To\_Off\_Supported* = NO
    - iii. The test stops here.
  - d. If the UUT responds with a *Not\_Supported* Message:
    - i. The check fails in PD2 mode.
    - ii. The Tester checks *VCONN\_Swap\_To\_Off\_Supported* = NO
    - iii. The check fails if the UUT ever sourced *VConn*
    - iv. The Test stops here.
5. The Tester sends a *PS\_RDY* Message.
6. The Tester checks that the UUT does not present *VCONN* at the non-CC line within *tVCONNSourceOff*. [TEST.PD.PROT.SRC.8#2]
7. If the UUT sends the *VCONN\_Swap* Message in the PD3 mode, then the Tester sends the *Accept* Message and go to step 9. Else, the Tester sends another *VCONN\_Swap*.
8. The Tester checks the response. [TEST.PD.PROT.SRC.8#3]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *VCONN\_Swap* every *tVCONNSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *VCONN\_Swap* for *nCtsWaitCount* times.
  - b. The check fails if the UUT responds with *Reject* Message and the VIF field *VCONN\_Swap\_To\_On\_Supported* = Yes. The test ends here.
  - c. The check fails if the UUT responds with *Not\_Supported*. The test ends here.
  - d. The check fails if the UUT responds with an *Accept* Message and the VIF field *VCONN\_Swap\_To\_On\_Supported* = No. The test ends here.
9. The Tester checks that the UUT sends a *PS\_RDY* Message within *tVCONNSourceOn*. [TEST.PD.PROT.SRC.8#4]
10. If the VIF field *Type\_C\_Sources\_VCONN* = NO, the Tester checks that *VCONN* is not present at the non-CC line. [TEST.PD.PROT.SRC.8#5]
11. If UUT sends the *VCONN\_Swap* Message, the Tester rejects the *VCONN\_Swap* Message and the Tester sends a *VCONN\_Swap* Message.
12. The Tester checks the response. [TEST.PD.PROT.SRC.8#6]
  - a. If the UUT responds with a *Wait* Message:
    - i. The check fails if the UUT responds with a *Wait* Message in PD3 mode and the Test stops here.
    - ii. If the UUT responds with a *Wait* Message in PD2 mode, the Tester resends *VCONN\_Swap* every *tVCONNSwapWait* + *tCtsWait* for



- nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *VCONN\_Swap* for *nCtsWaitCount* times in PD2 mode.
- b. If the UUT responds with an *Accept* Message, the Tester checks that *VCONN\_Swap\_To\_Off\_Supported* = YES
  - c. If the UUT responds with a *Reject* Message:
    - i. The check fails in PD3 mode.
    - ii. The Tester checks *VCONN\_Swap\_To\_Off\_Supported* = NO
    - iii. The test stops here.
  - d. If the UUT responds with a *Not\_Supported* Message:
    - i. The check fails in PD2 mode.
    - ii. The Tester checks *VCONN\_Swap\_To\_Off\_Supported* = NO
    - iii. The check fails if the UUT ever sourced *Vconn*.
    - iv. The Test stops here.
13. The Tester holds off sending a *PS\_RDY* Message and checks the response. [TEST.PD.PROT.SRC.8#7]
- a. The check fails if UUT does not send a *Hard Reset* signal after the max of *tVCONNSourceOn* in the PD2 mode.
  - b. The check fails if UUT does not send a *Hard Reset* signal between the min and max of *tVCONNSourceTimeout* in the PD3 mode.

#### TEST.PD.PROT.SRC.9 *PR\_Swap* Response

**Description:** The Tester verifies that the initial Source UUT responds correctly to the *PR\_Swap* Message.

#### Test Specific Tester Behavior:

- The Tester is a Dual-Role Power, initially a Sink.
- The Tester sends *Sink\_Capabilities* setting B27 (Unconstrained Power) to 1b COMMON.PROC.PD.3

#### Test Conditions:

	Consumer Only	Provider Only, C/P, P/C, DRP
Rev2Src		✓
Rev3ChkdSrc		✓

#### Test Procedures:

2. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field *PD\_Port\_Type* set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
3. The Tester sends a *PR\_Swap* Message.
4. The Tester checks the response. [TEST.PD.PROT.SRC.9#1]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *PR\_Swap* every *tPRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. Once the Tester has resent *PR\_Swap* for *nCtsWaitCount* times, the check fails, and the test ends here.



- b. If the UUT responds with *Reject* or *Not\_Supported*, check that the VIF field PD\_Port\_Type is set to “Provider Only” or VIF field Accepts\_PR\_Swap\_As\_Src = No. The test ends here.
  - c. The check fails if the UUT responds with an *Accept* Message and the VIF field PD\_Port\_Type is not set to either “Provider/Consumer”, “Consumer/Provider”, or “DRP”. The test ends here.
  - d. The check fails if the UUT responds with an *Accept* Message and the VIF field Accepts\_PR\_Swap\_As\_Src = No
5. The Tester draws no more than *iSwapStandby* within *tSrcTransition* min after it has sent the last *GoodCRC*.
6. The Tester checks that the UUT asserts Rd and sends a *PS\_RDY* Message after the UUT driving VBUS voltage to *vSafe0V*. [TEST.PD.PROT.SRC.9#2]
7. The Tester checks that the UUT sends *PS\_RDY* within *tPSSourceOff* min after the last bit of *GoodCRC* was sent in response to the received *Accept*. [TEST.PD.PROT.SRC.9#3]
8. The Tester presents *vSafe5V* on VBUS and then it sends a *PS\_RDY* Message, both within *tNewSRC* max of the last *GoodCRC* sent by the UUT.
9. The Tester transmits *Source\_Capabilities* Message with single PDO (5V @ 500mA, B29 Dual-Role Power set to 1b). The Tester checks that the UUT sends a *Request* Message. [TEST.PD.PROT.SRC.9#4] The Tester uses the common procedure COMMON.PROC.PD.10 to establish a PD contract.
10. The UUT is now a Sink, and the Tester is a Source.
11. The Tester sends a *PR\_Swap* Message.
12. The Tester checks the response. [TEST.PD.PROT.SRC.9#5]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *PR\_Swap* every *tPRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. Once the Tester has resent *PR\_Swap* for *nCtsWaitCount* times, the check fails, and the test ends here.
  - b. The check fails if the UUT responds with a *Reject* Message and that the VIF field Accepts\_PR\_Swap\_As\_Snk = Yes.
  - c. The check fails if the UUT responds with an *Accept* Message and that the VIF field Accepts\_PR\_Swap\_As\_Snk = No.
  - d. The check fails if the UUT response with a *Not\_Supported* Message.
13. At *tSrcTransition* max (this delay is from the last bit of *GoodCRC* sent in response to the *Accept*), the Tester drives the VBUS voltage to *vSafe0V* within *tSrcSwapStdbby* max.
14. The Tester sends a *PS\_RDY* at the deadline limit of *tPSSourceOff* min, the delay is from the time the last bit of the EOP of the *GoodCRC* corresponding to the *Accept* Message.
15. The tester checks that UUT asserts Rp before the UUT has applied *vSafe5V* [TEST.PD.PROT.SRC.9#6]
16. The Tester checks that the UUT sends *PS\_RDY* only after the UUT has applied *vSafe5V*. [TEST.PD.PROT.SRC.9#7]
17. The Tester checks that the *PS\_RDY* from the UUT is within *tPSSourceOn* min. [TEST.PD.PROT.SRC.9#8] This delay is from the *GoodCRC* sent by the UUT in response to the *PS\_RDY* sent by the Tester.
18. The Tester checks that the UUT sends a *Source\_Capabilities* Message after *tSwapSourceStart* and before *tFirstSourceCap* max. [TEST.PD.PROT.SRC.9#9].
19. The Tester waits for 5 second period to check correct usage of Rp, and the Tester replies any message sent from the UUT with a proper response.

#### TEST.PD.PROT.SRC.10 PR\_Swap – PSSourceOnTimer Timeout

**Description:** As a new Source, the Tester intentionally does not send a *PS\_RDY* Message after a *PR\_Swap* in order to force a PSSourceOnTimer timeout on the UUT and verifies it is correctly implemented.

#### Test Specific Tester Behavior:

- The Tester is a Dual-Role Power, initially a Sink.
- The Tester sends *Sink\_Capabilities* setting B27 (Unconstrained Power) to 1b COMMON.PROC.PD.3

#### Test Conditions:

	Consumer Only	Provider Only, C/P, P/C, DRP
Rev2Src		✓
Rev3ChkdSrc		✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester sends a *PR\_Swap* Message.
3. The Tester checks the response. [TEST.PD.PROT.SRC.10#1]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *PR\_Swap* every  $tPRSwapWait + tCtsWait$  for  $nCtsWaitCount$  times. Once the Tester has resent *PR\_Swap* for  $nCtsWaitCount$  times, the check fails, and the test ends here.
  - b. If the UUT responds with *Reject* or *Not\_Supported*, check that the VIF field PD\_Port\_Type is set to “Provider only” or VIF field Accepts\_PR\_Swap\_As\_Src is set to No. The test ends here.
  - c. The check fails if the UUT responds with an *Accept* Message and the VIF field PD\_Port\_Type is not set to either “Provider/Consumer”, “Consumer/Provider”, or “DRP”. The test ends here.
4. The Tester draws no more than *iSwapStandby* within  $tSrcTransition$  min after it has sent the last *GoodCRC*.
5. The Tester checks the response: [TEST.PD.PROT.SRC.10#2]
  - a. The check fails if the UUT does not send a *PS\_RDY* Message after the UUT driving  $V_{BUS}$  voltage to *vSafe0V*.
  - b. The check fails if the UUT does not send *PS\_RDY* within  $tSrcTransition$  max plus  $tSrcSwapStdby$  max after the last bit of *GoodCRC* was sent in response to the received *Accept*.
6. The Tester presents *vSafe5V* on  $V_{BUS}$  but does not send a *PS\_RDY* Message, in order to force a *PSSourceOnTimer* timeout on the UUT.
7. A USB Type-C Error Recovery is expected from the UUT. The Tester checks the response: [TEST.PD.PROT.SRC.10#3]

- a. The check fails if a USB Type-C Error Recovery is detected before *tPSSourceOn* min after the time the last bit of the EOP of the *GoodCRC* Message corresponding to the received *PS\_RDY* Message.
- b. The check fails if a USB Type-C Error Recovery is not detected within *tPSSourceOn* max after the last bit of the EOP of the *GoodCRC* Message corresponding to the received *PS\_RDY* Message.

#### *TEST.PD.PROT.SRC.11 Unexpected Message Received in Ready State*

**Description:** As a Sink, the Tester sends an unexpected message to the UUT after setting up a contract with the UUT and verifies the UUT does *Soft Reset*.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
Rev2Src		✓
Rev3ChkdSrc		✓

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester sends an *Accept* Message to the UUT.
3. The check fails if a *Soft Reset* Message is not received within *tProtErrSoftReset* max. [TEST.PD.PROT.SRC.11#1] This delay is measured from the EOP of the *GoodCRC* corresponding to the *Accept* Message has been received to the time the EOP the *Soft Reset* Message has been received.

#### *TEST.PD.PROT.SRC.12 Get\_Sink\_Cap Response*

**Description:** The Tester verifies that the Source UUT responds correctly to *Get\_Sink\_Cap* Message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
Rev2Src		✓
Rev3ChkdSrc		✓

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.

- b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester sends a *Get\_Sink\_Cap* Message to the UUT.
3. The Tester checks the response. [TEST.PD.PROT.SRC.12#1] /\*same as TEST.PD.PROT.ALL.4 12.c \*/
  - a. If the VIF field PD\_Port\_Type is set to Provider/Consumer, check that the UUT sends a *Sink\_Capabilities* Message.
  - b. In PD2 mode, if the VIF field PD\_Port\_Type is set to Provider only, check that the UUT sends a *Reject* Message.
  - c. In PD3 mode, if the VIF field PD\_Port\_Type is set to Provider only, check that the UUT sends a *Not\_Supported* Message.

#### TEST.PD.PROT.SRC.13 PR\_Swap – GoodCRC not sent in Response to PS\_RDY

Description: As a new Source, the Tester intentionally does not reply *GoodCRC* Message to the *PS\_RDY* Message after a *PR\_Swap* in order to force the UUT to do an Error Recovery and verifies it is correctly implemented.

#### Test Specific Tester Behavior:

- The Tester is a Dual-Role Power, initially a Sink.
- The Tester sends *Sink\_Capabilities* setting B27 (Unconstrained Power) to 1b COMMON.PROC.PD.3

#### Test Conditions:

	Consumer Only	Provider Only, C/P, P/C, DRP
Rev2Src		✓
Rev3ChkdSrc		✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester sends a *PR\_Swap* Message.
3. The Tester checks the response. [TEST.PD.PROT.SRC.13#1]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *PR\_Swap* every  $tPRSwapWait + tCtsWait$  for  $nCtsWaitCount$  times. Once the Tester has resent *PR\_Swap* for  $nCtsWaitCount$  times, the check fails, and the test ends here.
  - b. If the UUT responds with *Reject* or *Not\_Supported*, check that the VIF field PD\_Port\_Type is set to “Provider Only” or VIF field Accepts\_PR\_Swap\_As\_Src is set to No. The test ends here.
  - c. The check fails if the UUT responds with an *Accept* Message and the VIF field PD\_Port\_Type is not set to either “Provider/Consumer”, “Consumer/Provider”, or “DRP”. The test ends here.
4. The Tester draws no more than  $iSwapStandby$  within  $tSrcTransition$  min after it has sent the last *GoodCRC*.

5. The Tester checks the response: [TEST.PD.PROT.SRC.13#2]
  - a. The check fails if the UUT does not send a *PS\_RDY* Message after the UUT driving  $V_{BUS}$  voltage to *vSafe0V*.
  - b. The check fails if the UUT does not send *PS\_RDY* within *tSrcTransition* max plus *tSrcSwapStdby* max after the last bit of *GoodCRC* was sent in response to the received *Accept*.
6. Upon receipt of the *PS\_RDY* Message, the Tester:
  - a. Intentionally does not respond with *GoodCRC*.
  - b. Applies *Rp* in place of *Rd*.
  - c. Does not Send further messages.
7. A USB Type-C Error Recovery is expected from the UUT. The check fails if a USB Type-C Error Recovery is not detected within *tCtsErrorRecovery* after the last bit of the EOP of the last retried *PS\_RDY* Message. [TEST.PD.PROT.SRC.13#3]

### 5.3.2 PD3 Mode

The tests in this subsection are run only in PD3 mode.

The tests in this subsection are applicable to Source capable UUT with VIF field *PD\_Port\_Type* set to Provider Only, Provider/Consumer and DRP.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (i.e., a Provider/Consumer or Consumer/Provider) The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### TEST.PD.PROT.SRC3.1 SourceCapabilityTimer Timeout

Description: The Tester waits for a *Source\_Capabilities* Message from the Source UUT. The Tester intentionally does not reply with a *GoodCRC* to force a SourceCapabilityTimer timeout, then verifies it is correctly implemented.

Test Specific Tester Behavior: Do not emulate the cable during this test.

Test Conditions:

	Consumer Only, C/P	Provider Only, DRP, P/C
<i>Rev3ChkdSrc</i>		✓

Test Procedures:

1. The test starts in a disconnected state.
2. The Tester applies *Rd* and it waits for  $V_{BUS}$  present.

3. The check fails if the first bit of a *Source\_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after  $V_{BUS}$  present. [TEST.PD.PROT.SRC3.1#1]
4. Upon receipt of the first *Source\_Capabilities* Message from the Source UUT, the Tester intentionally does not reply with a *GoodCRC* Message and waits for the next *Source\_Capabilities* Message. The Tester does not reply with a *GoodCRC* Message for the duration of this test.
5. In the following steps the Tester measures the delay from the last bit of a *Source\_Capabilities* Message EOP to the first bit of the preamble of the following *Source\_Capabilities* Message received. [TEST.PD.PROT.SRC3.1#2]
  - a. The Tester checks that the time between the first *Source\_Capabilities* received, and the second *Source\_Capabilities* received is within *tReceive* max + *tRetry*.
  - b. The Tester checks that the time between the second *Source\_Capabilities* received, and the third *Source\_Capabilities* received is within *tReceive* max + *tRetry*.
  - c. The Tester measures the time between the third *Source\_Capabilities* received and the fourth *Source\_Capabilities* received.
    - i. The check fails if this time is larger than *tTypeCSendSourceCap* max + *tReceive* max (201.1ms)
    - ii. The check fails if this time is smaller than *tTypeCSendSourceCap* min + *tReceive* min (100.9ms)
  - d. The Tester checks that the time between the fourth *Source\_Capabilities* received, and the fifth *Source\_Capabilities* received is within *tReceive* max + *tRetry*.

#### TEST.PD.PROT.SRC3.2 SenderResponseTimer Timeout

**Description:** As a Sink, the Tester intentionally does not send the *Request* Message to force a SenderResponseTimer timeout on the Source UUT. The Tester verifies correct implementation of this timer.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, C/P	Provider Only, DRP, P/C
<i>Rev3ChkdSrc</i>		✓

**Test Procedures:**

1. The test starts in a disconnected state.
2. The Tester applies  $R_d$ , and it waits for  $V_{BUS}$  for *tNoResponse* max.
3. The check fails if the first bit of a *Source\_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after  $V_{BUS}$  present. [TEST.PD.PROT.SRC3.2#1]
4. The Tester intentionally does not send a *Request* Message and waits for a *Hard Reset*.
5. The Tester checks that a *Hard Reset* is completed between *tSenderResponse* min and *tSenderResponse* max + *tSendHardReset* max (see Table 19 Timing Table & Calculations), the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the last bit of *Hard Reset* EOP has been received. [TEST.PD.PROT.SRC3.2#2]



### TEST.PD.PROT.SRC3.3 Get\_Source\_Cap\_Extended Response

**Description:** As a Sink, the Tester verifies that the Source UUT responds correctly to *Get\_Source\_Cap\_Extended* Message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
<i>Rev3ChkdSrc</i>		✓
<i>Rev3UnchkdSrc</i>		✓

**Test Procedures:**

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
- The Tester runs checks for the VIF parameter Has\_Invariant\_PDOs and *Source Capabilities*:
  - If VIF parameter Has\_Invariant\_PDOs is set to N and the current *Source Capabilities* do not match those in the VIF, run common procedure COMMON.PROC.PD.18, then the test proceeds to the next step.
  - Otherwise, the test proceeds to the next step.
- The Tester sends a *Get\_Source\_Cap\_Extended* Message to the UUT.
- The check fails if the UUT does not send either *Not\_Supported* or *Source\_Capabilities\_Extended* Message. [TEST.PD.PROT.SRC3.3#1]

### TEST.PD.PROT.SRC3.4 Alert Response Source Input Change

**Description:** As a Sink, the Tester sends *Alert* Message indicating Source Input Change and verifies the response messages. Note that there isn't any specific failure condition for this test, however, failure as a side effect of this will be captured as part of the COMMON checks and procedures.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
<i>Rev3ChkdSrc</i>		✓
<i>Rev3UnchkdSrc</i>		✓

**Test Procedures:**

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.



- b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester sends an *Alert* Message to the UUT with Type of *Alert* field = Source Input Change Event (Bit 5).
3. The test passes if one of the following conditions happens [TEST.PD.PROT.SRC3.4#1]:
  - a. A *Not\_Supported* Message is received as a response to the last *Alert* Message and the UUT as a Source has no source PPS APDO in VIF.
  - b. A *Get\_Status* Message is received within *tSenderResponse* min after the last *Alert* Message was sent.
  - c. No message is received for *tSenderResponse* min.

**NOTE:** *Alert Shall* be supported when *SPR PPS Mode* is supported

	Not_supported	Get_status	No message
Source APDO in VIF	Fail	Pass	Pass
No APDO in VIF	Pass	Pass	Pass

#### TEST.PD.PROT.SRC3.5 Alert Response Battery Status Change

**Description:** As a Sink, the Tester sends *Alert* Message indicating Battery Status Change and verifies the response messages.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
<i>Rev3ChkdSrc</i>		✓
<i>Rev3UnchkdSrc</i>		✓

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester sends an *Alert* Message to the UUT
  - a. Type of *Alert* field = Battery Status Change Event (Bit 1)
  - b. Fixed Batteries = Battery 0 (B20)
3. The test passes if one of the following conditions happens[TEST.PD.PROT.SRC3.5#2]:
  - a. A *Not\_Supported* Message is received as a response to the last *Alert* Message and the UUT as a Source has no source PPS APDO in VIF.
  - b. A *Get\_Status* Message is not received within *tCtsAfterLastPoint* after the last *Alert* Message was sent.

Otherwise, the test fails and it stops here.

**NOTE:** *Alert Shall* be supported when *SPR PPS Mode* is supported.

	Not_supported	Get_battery_status	No message
--	---------------	--------------------	------------

Source APDO in VIF	Fail	Pass	Pass
No APDO in VIF	Pass	Pass	Pass

#### TEST.PD.PROT.SRC3.6 Soft Reset Sent when SinkTxOK

**Description:** As a Sink, the Tester forces the UUT to send *Soft Reset* and verifies *Soft Reset* is sent when the Rp value is *SinkTxOK*.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
<i>Rev3ChkdSrc</i>		✓

**Test Procedures:**

4. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
5. The Tester sends a *Get\_Source\_Cap* Message to the UUT.
6. Upon receipt of the *Source\_Capabilities* Message, the Tester doesn't reply with *GoodCRC*.
7. The check fails if a *Soft Reset* Message is not sent by the UUT within *tReceive* max + *tSoftReset* max, the delay is measured from the time the last bit of the last retransmitted *Source\_Capabilities* Message EOP has been received to the time the last bit of the *Soft Reset* Message EOP has been received. [TEST.PD.PROT.SRC3.6#1]

#### TEST.PD.PROT.SRC3.7 Get\_PPS\_Status Response

**Description:** The Tester verifies that the Source UUT responds correctly to Get\_PPS\_Status message.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
<i>Rev3ChkdSrc</i>		✓
<i>Rev3UnchkdSrc</i>		✓

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the Consumer-Provider as Source UUT PPS COMMON.PROC.BU.9.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source for PPS test COMMON.PROC.BU.4.

2. The Tester sends a Get\_PPS\_Status message to the UUT.
3. The check fails if the Tester does not receive PPS\_Status message.  
[TEST.PD.PROT.SRC3.7#1]
4. The Tester runs the following additional field checks for the PPSSDB:  
[TEST.PD.PROT.SRC3.7#2]
  - a. Output Voltage field shall be within  $\pm 3\%$  of the actual output voltage rounded to the nearest 20mV or 0xFFFF.
  - b. Output Current field shall be 0x00 to 0x03(inclusive) or 0xFF.
  - c. For Real Time Flags field
  - d. Bit 0 shall be 0
  - e. PTF shall be 0x00 (Not Supported) or 0x01 (Normal)
  - f. OMF shall be 0 (Constant Voltage mode)
  - g. Bits 4..7 shall be 0

#### TEST.PD.PROT.SRC3.8 SourcePPSCommTimer Deadline

**Description:** As a Sink, the Tester sends the second *Request* Message after *tPPSRequest* max, verifies *SourcePPSCommTimer* is not timeout at the UUT, and the timer is correctly reinitialized.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
<i>Rev3ChkdSrc</i>		✓

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the Consumer-Provider as Source UUT PPS COMMON.PROC.BU.9.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source for PPS test COMMON.PROC.BU.4.
2. The check fails if a *Hard Reset* is received within *tPPSRequest* max.  
[TEST.PD.PROT.SRC3.8#1] The delay is measured from the time the last bit of the EOP of the *GoodCRC* Message sent by the UUT in response to the previous *Request* Message to the time the first bit of the *Hard Reset* is received.
3. The Tester sends the second *Request* Message to request the Programmable Power Supply APDO at 5.5V 1A, *tPPSRequest* max after the previous *Request* Message. The check fails if the Tester cannot send the second *Request* Message at *tPPSRequest* max because the UUT has presented *SinkTxNG*. [TEST.PD.PROT.SRC3.8#2]
4. The check fails if the UUT does not respond with either *Accept*, *Reject* or *Wait* Message, and the test stops here. If the *Reject* Message received, the Tester reports check with a warning, and the test stops here. [TEST.PD.PROT.SRC3.8#5]
5. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until *Accept* Message is received. The check

fails if the UUT responds with Wait Message after *nCtsWaitCount* retries, and the test stops here. [TEST.PD.PROT.SRC3.8#6]

6. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [TEST.PD.PROT.SRC3.8#3]
7. The check fails if a *Hard Reset* is received within *tPPSRequest* max. [TEST.PD.PROT.SRC3.8#4] The delay is measured from the time the last bit of the EOP of the *GoodCRC* Message sent by the UUT in response to the second *Request* Message to the time the first bit of the *Hard Reset* is received.

#### TEST.PD.PROT.SRC3.9 SourcePPSCommTimer Timeout

**Description:** As a Sink PPS, the Tester intentionally stops resending *Request* Message, in order to force a SourcePPSCommTimer timeout on the UUT and verifies it is correctly implemented.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
<i>Rev3ChkdSrc</i>		✓

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the Consumer-Provider as Source UUT PPS COMMON.PROC.BU.9.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source for PPS test COMMON.PROC.BU.4.
2. The Tester repeats the following steps 3 times:
  - a. At *tPPSRequest* max after the previously sent *Request* Message, the Tester sends a *Request* Message requesting Programmable Power Supply APDO 5.5V 1A. The check fails if the Tester cannot send the *Request* Message at *tPPSRequest* max because the UUT has presented *SinkTxNG*. [TEST.PD.PROT.SRC3.9#1]
3. The Tester intentionally stops sending *Request* Message.
4. The check fails if a *Hard Reset* is received before *tPPSTimeout* min, or not received within *tPPSTimeout* max. [TEST.PD.PROT.SRC3.9#2] The delay is measured from the time the last bit of the EOP of the *GoodCRC* Message sent by the UUT in response to the previous *Request* Message to the time the first bit of the preamble of the *Hard Reset* is received.

#### TEST.PD.PROT.SRC3.10 SourcePPSCommTimer Stopped

**Description:** As a Sink, the Tester request a PPS APDO first and then request a fixed PDO, verifies SourcePPSCommTimer is stopped at the UUT.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
<i>Rev3ChkdSrc</i>		✓

#### Test Procedures:

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the Consumer-Provider as Source UUT PPS COMMON.PROC.BU.9.
  - The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source for PPS test COMMON.PROC.BU.4.
- The check fails if a *Hard Reset* is received within tPPSRequest max.  
[TEST.PD.PROT.SRC3.10#1] The delay is measured from the time the last bit of the EOP of the *GoodCRC* Message sent by the UUT in response to the previous *Request* Message to the time the first bit of the *Hard Reset* is received.
- The Tester sends the second *Request* Message to request the Fixed Supply PDO at 5V 0.1A, tPPSRequest max after the previous *Request* Message. The check fails if the Tester cannot send the second *Request* Message at tPPSRequest max because the UUT has presented *SinkTxNG*. [TEST.PD.PROT.SRC3.10#2]
- The check fails if the UUT does not respond with either *Accept*, *Reject* or *Wait* Message, and the test stops here. If the *Reject* Message received, the Tester reports the check with a warning, and the test stops here. [TEST.PD.PROT.SRC3.10#3]
- If the UUT response with *Wait* Message, the Tester resends *Request* Message every tSinkRequest for nCtsWaitCount times or until *Accept* Message is received. The check fails if the UUT responds with Wait Message after nCtsWaitCount retries, and the test stops here. [TEST.PD.PROT.SRC3.10#5]
- The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within tSrcTransReq from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [TEST.PD.PROT.SRC3.10#6]
- The check fails if a *Hard Reset* is received within tPPSTimeout max + tCtsHardReset. [TEST.PD.PROT.SRC3.10#4] The delay is measured from the time the last bit of the EOP of the *GoodCRC* Message sent by the UUT in response to the second *Request* Message to the time the first bit of the *Hard Reset* is received.

#### *TEST.PD.PROT.SRC3.11 GoodCRC Specification Revision Compatibility*

Description: As a Sink, the Tester verifies the UUT accepts *GoodCRC* with Specification Revision set to 00b, 01b, or 10b.

Test Specific Tester Behavior: N/A

#### Test Conditions:

	Consumer Only, C/P	Provider Only, DRP, P/C
<i>Rev3ChkdSrc</i>		✓

#### Test Procedures:

1. The test starts in a disconnected state.
2. The Tester applies Rd, and it waits for V<sub>BUS</sub> for *tNoResponse* max.
3. The check fails if the first bit of a *Source\_Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after V<sub>BUS</sub> present. [TEST.PD.PROT.SRC3.11#1]
4. The Tester sends a *GoodCRC* (acknowledging the *Source\_Capabilities* Message) with Specification Revision set to 00b. The check fails if the UUT retransmits the *Source\_Capabilities* Message. [TEST.PD.PROT.SRC3.11#2]
5. Repeat the test with Specification Revision in *GoodCRC* set to 01b and 10b.

#### TEST.PD.PROT.SRC3.12 FR\_Swap Without Signaling

**Description:** The tester checks the UUT response to an invalid *FR\_Swap* Message

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, DRP, P/C, C/P
<i>Rev3ChkdSrc</i>		✓

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester sends a *FR\_Swap* Message to the UUT.
3. The Tester checks the response: [TEST.PD.PROT.SRC3.12#1]
  - a. If FR\_Swap\_Reqd\_Type\_C\_Current\_As\_Initial\_Source field (in the VIF) is zero value, the Tester checks that the UUT sends *Not\_Supported*.
  - b. If FR\_Swap\_Reqd\_Type\_C\_Current\_As\_Initial\_Source field (in the VIF) is nonzero value, the Tester checks that that PD\_Port\_Type field (in the VIF) is not set to Provider Only. The check fails if a *Hard\_Reset* is not received within *tProtErrHardReset* max. The delay is measured from the time the last bit of the EOP of the *GoodCRC* Message sent by the UUT in response to the *FR\_Swap* Message to the time the first bit of the *Hard\_Reset* is received.

#### TEST.PD.PROT.SRC3.13 Cable Type Detection

**Description:** As a Sink, the Tester verifies that the UUT does cable detection if it offers currents in excess of 3A, voltages in excess of 20V or both.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

The test is applicable if VIF field Captive\_Cable = NO

	Consumer Only, C/P	Provider Only, DRP, P/C
--	--------------------	-------------------------



Test Procedures:

1. The test starts in a disconnected state.
2. The Tester applies Rd and Ra, and it waits for V<sub>BUS</sub> present.
3. The check fails if the first bit of a *Source Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after V<sub>BUS</sub> present. [TEST.PD.PROT.SRC3.13#4]
4. The Tester runs checks for the VIF parameter Has\_Invariant\_PDOs and *Source Capabilities*:
  - a. If VIF parameter Has\_Invariant\_PDOs is set to N and the current *Source Capabilities* do not match those in the VIF, run common procedure COMMON.PROC.PD.18, then the test proceeds to the next step.
  - b. Otherwise, the test proceeds to the next step.
5. The Tester checks the response:
  - a. If a SOP' *Discover Identity* Command is received from the UUT, the Tester replies as a Passive Cable with V<sub>BUS</sub> Current Handling Capability set to 5A and Maximum V<sub>BUS</sub> Voltage set to 50V.
  - b. Without receiving a SOP' *Discover Identity* Command from the UUT, the check fails if the UUT sends *Source Capabilities* offering currents in excess of 3A and/or voltages in excess of 20V. The test stops here. [TEST.PD.PROT.SRC3.13#1]
6. The Tester removes all terminations and simulates a disconnection
7. The Tester applies Rd and Ra, and it waits for V<sub>BUS</sub> present.
8. The check fails if the first bit of a *Source Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after V<sub>BUS</sub> present. [TEST.PD.PROT.SRC3.13#5]
9. The Tester runs checks for the VIF parameter Has\_Invariant\_PDOs and *Source Capabilities*:
  - a. If VIF parameter Has\_Invariant\_PDOs is set to N and the current *Source Capabilities* do not match those in the VIF, run common procedure COMMON.PROC.PD.18, then the test proceeds to the next step.
  - b. Otherwise, the test proceeds to the next step.
10. The Tester checks the response:
  - a. If a SOP' *Discover Identity* Command is received, the Tester replies as a Passive Cable with V<sub>BUS</sub> Current Handling Capability set to 3A and Maximum V<sub>BUS</sub> Voltage set to 20V.
  - b. The check fails if the UUT sends *Source Capabilities* offering currents in excess of 3A and/or voltages in excess of 20V. [TEST.PD.PROT.SRC3.13#2]
11. The Tester removes all terminations and simulates a disconnection
12. The Tester applies Rd but does not present Ra, and it waits for V<sub>BUS</sub> present.
13. The check fails if the first bit of a *Source Capabilities* Message is not received from the UUT within *tFirstSourceCap* max after V<sub>BUS</sub> present. [TEST.PD.PROT.SRC3.13#6].
14. The Tester runs checks for the VIF parameter Has\_Invariant\_PDOs and *Source Capabilities*:
  - a. If VIF parameter Has\_Invariant\_PDOs is set to N and the current *Source Capabilities* do not match those in the VIF, run common procedure COMMON.PROC.PD.18, then the test proceeds to the next step.



- b. Otherwise, the test proceeds to the next step.
15. The Tester shall disregard *Discover Identity* SOP'messages if any.
16. The Tester checks the UUT Source Capability response. The check fails if the UUT sends *Source\_Capabilities* offering currents in excess of 3A and/or voltages in excess of 20V [TEST.PD.PROT.SRC3.13#3].

#### TEST.PD.PROT.SRC3.14 Source Info

Description: As a Sink, the Tester verifies that the UUT replies to the *Get\_Source\_Info* Message with Source\_Info.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only	Provider Only, C/P, P/C, DRP
<i>Rev3ChkdSrc</i>		✓
<i>Rev3UnchkdSrc</i>		✓

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester sends the *Get\_Source\_Info* Message to the UUT.
3. The Tester checks the response of the *Get\_Source\_Info* [TEST.PD.PROT.SRC3.14#2]:
  - a. If the UUT responds with a *Not\_Supported* Message, the Tester checks Has\_Invariant\_PDOs VIF field is set to Yes and EPR\_Supported\_As\_Src VIF field is set to No and check number of SRC+DRP VIF objects is equal to 1 (Use Type\_C\_State\_Machine VIF field to find the number of SRC/DRP VIF objects) and the test stops here.
  - b. Else, the check fails if the UUT does not reply with the *Source\_Info* Message, and the test stops here.
4. The Tester checks the response, and the check fails if one of the following conditions is met: [TEST.PD.PROT.SRC3.14#1]
  - a. The Number of Data Objects in the *Source\_Info* Message is not 2.
  - b. In SIDO1 and SIDO2, B31 (Port Type) does not match Port\_Managed\_Guaranteed\_Type VIF field.
  - c. In SIDO2, if B30 (DPS Port) is 1, Bit31 (Port Type) is not 0.
  - d. In SIDO2, B30 (DPS Port) does not match to the DPS\_Supported VIF field.
  - e. In SIDO1, B30...24 are not Zeros.
  - f. In SIDO2, B29...18 are not Zeros.
  - g. In SIDO1, B23...16 (Port Maximum PDP) is not equal to the round down value of PD\_Power\_As\_Source VIF field/1000.
  - h. In SIDO1, B15...8 (Port Present PDP) is higher than Maximum PDP

- i. In SIDO1, B7...0 (Port Reported PDP) is higher than Maximum PDP or not within the computed PDP range of Source Capabilities from the current Explicit Contract. The PDP of the Source Capabilities computed as the largest of the products of each fixed PDO voltage times its maximum current returned in the *Source\_Capabilities* Message.
    - $PDP\_N = (\text{Maximum Current, B9...0}) * (\text{Voltage, B19...10}) * 10^{-6}$  from each fixed PDO where N= PDO number(1...N)
    - $PDP\_Max = \text{Max}(PDP\_1, PDP\_2, \dots PDP\_N)$
    - $PDP\_Max\_RndDn = \text{Integer of Rounded down } PDP\_Max \text{ to the nearest } 1/10^{\text{th}}$
    - $PDP\_Max\_RndUp = \text{Integer of Rounded up } PDP\_Max \text{ to the nearest } 1/10^{\text{th}}$
    - $PDP\_Max\_RndDn \leq \text{Port Reported PDP} \leq PDP\_Max\_RndUp$
  - j. In SIDO2, B8...0 (Port Guaranteed PDP) does not match the PD\_Power\_As\_Source\_Guaranteed VIF field
  - k. If the Port\_Managed\_Guaranteed\_Type VIF field is set to 1 (Guaranteed), and
    1. In SIDO1, B23...16 (Port Maximum PDP) is not equal to B15...B8 (Port Present PDP)
    2. In SIDO2, B17...9 (Port Maximum PDP) is not equal to B8...0 (Port Guaranteed PDP)
    3. If the PDP of the Source Capabilities from the current *Explicit Contract* is less than SIDO B8...0 (Port Guaranteed PDP)
5. The Tester disconnects from the UUT and waits for *tV<sub>BUS</sub>OFF* (650ms, Type-C spec).
6. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1
  - c. During bring-up, if SOP' *Discover Identity* Command is received, the Tester replies as Passive Cable with V<sub>BUS</sub> Current Handling Capability set to 10b(5A), Maximum V<sub>BUS</sub> Voltage set to 00b(20V) and EPR Mode Capable Field set to 0b (EPR Mode Capable)
7. The Tester sends the *Get\_Source\_Info* to the UUT.
8. The Tester checks the response of *Get\_Source\_info* [TEST.PD.PROT.SRC3.14#3] :
  - a. If the UUT responds with a *Not\_Supported* Message, the Tester checks Has\_Invariant\_PDOs VIF field is set to Yes and EPR\_Supported\_As\_Src VIF field is set to No, and check number of SRC+DRP VIF objects is equal to 1 (Use Type\_C\_State\_Machine VIF field to find the number of SRC/DRP VIF objects) and stops here.
  - b. Else, the check fails if the UUT does not reply with the *Source\_Info* Message and the test stops here.
9. The Tester checks the response, and the check fails if one of the following conditions is met in the *Source\_Info* Message [TEST.PD.PROT.SRC3.14#4]:
  - a. The Number of Data Objects in the *Source\_Info* Message is not 2.

- b. In SDO1 and SDO2, B31 (Port Type) does not match Port\_Managed\_Guaranteed\_Type VIF field.
  - c. In SDO2, if B30 (DPS Port) in SDO2 is 1, Bit31 (Port Type) is not 0.
  - d. In SDO2, B30 (DPS Port) does not match to the DPS\_Supported VIF field
  - e. In SDO1, B30...24 are not Zeros.
  - f. In SDO2, B29...18 are not Zeros.
  - g. In SDO1, B23...16 (Port Maximum PDP) is not equal to the round down value of PD\_Power\_As\_Source VIF field/1000
  - h. In SDO1, B15...8 (Port Present PDP) is higher than Maximum PDP
  - i. In SDO1, B7...0 (Port Reported PDP) is higher than Maximum PDP or not within the computed PDP range of Source Capabilities from the current Explicit Contract. The PDP of the Source Capabilities computed as the largest of the products of each fixed PDO voltage times its maximum current returned in the *Source\_Capabilities* Message.
    - $PDP\_N = (Maximum\ Current, B9...0) * (Voltage, B19...10) * 10^{-6}$  from each fixed PDO where N= PDO number(1...N)
    - $PDP\_Max = Max(PDP\_1, PDP\_2, ...PDP\_N)$
    - $PDP\_Max\_RndDn = \text{Integer of Rounded down } PDP\_Max \text{ to the nearest } 1/10^{th}$
    - $PDP\_Max\_RndUp = \text{Integer of Rounded up } PDP\_Max \text{ to the nearest } 1/10^{th}$
    - $PDP\_Max\_RndDn \leq \text{Port Reported PDP} \leq PDP\_Max\_RndUp$
  - j. In SDO2, B8...0 (Port Guaranteed PDP) does not match PD\_Power\_As\_Source\_Guaranteed VIF field.
  - k. If VIF field Captive\_Cable = No, the Port\_Managed\_Guaranteed\_Type VIF field is set to 1 (Guaranteed) and EPR\_Supported\_As\_SRC is Yes, B23..16 (Port Maximum PDP) is equal to B15..8 (Port Present PDP) in SDO1.
  - l. If VIF field Captive\_Cable = YES and the Port\_Managed\_Guaranteed\_Type VIF field is set to 1 (Guaranteed) and EPR\_Supported\_As\_SRC is Yes, and B23..16 (Port Maximum PDP) is not equal to B15..8 (Port Present PDP) in SDO1
  - m. If VIF field Captive\_Cable = No, the PD\_Power\_as\_Source VIF field is greater than 100W and the Port\_Present in SDO1 is higher than 100W.
10. If VIF field is Captive\_Cable = Yes, the test stops here.
  11. The Tester disconnects from the UUT and waits for *tV<sub>BUSOFF</sub>*.
  12. There are 2 possible bring-up procedures:
    - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
    - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1
    - c. During bring-up, if the SOP' *Discover Identity* Command is received, the Tester intentionally does not reply with the *GoodCRC* Message.
  13. The Tester sends the *Get\_Source\_Info* Message to the UUT.
  14. The Tester checks the response of the *Get\_Source\_Info* Message [TEST.PD.PROT.SRC3.14#5].

- a. If the UUT responds with a Not\_Supported Message, the Tester checks Has\_Invariant\_PDOs VIF field is set to Yes and EPR\_Supported\_As\_Src VIF field is set to No and check number of SRC+DRP VIF objects is equal to 1 (Use Type\_C\_State\_Machine VIF field to find the number of SRC/DRP VIF objects) and test stops here.
  - b. Else, the check fails if the UUT does not reply with the *Source\_Info* Message and the test stops here.
15. The Tester checks the response, and the check fails if one of the following conditions is met in the *Source\_Info* Message [TEST.PD.PROT.SRC3.14#6].
  - a. The number of Data Objects in the Source\_Info Message is not 2.
  - b. In SIDO1 and SIDO2, B31 (Port Type) does not match Port\_Managed\_Guaranteed\_Type VIF field. If B30 (DPS Port) in SIDO2 is 1, Bit31 (Port Type) is not 0.
  - c. In SIDO2, B30 (DPS Port) does not match the DPS\_Supported VIF field.
  - d. In SIDO1, B30...24 are not Zeros.
  - e. In SIDO2, B29...18 are not Zeros.
  - f. In SIDO1, B23...16 (Port Maximum PDP) is not equal to the round down value of PD\_Power\_As\_Source VIF field/1000.
  - g. In SIDO1, B15...8 (Port Present PDP) is higher than Maximum PDP.
  - h. In SIDO1, B7...0 (Port Reported PDP) is higher than Maximum PDP or not within the computed PDP range of Source Capabilities from the current Explicit Contract. The PDP of the Source Capabilities computed as the largest of the products of each fixed PDO voltage times its maximum current returned in the *Source\_Capabilities* Message.
    - $PDP\_N = (Maximum\ Current, B9...0) * (Voltage, B19...10) * 10^{-6}$  from each fixed PDO where N= PDO number(1...N)
    - $PDP\_Max = Max(PDP\_1, PDP\_2, ...PDP\_N)$
    - $PDP\_Max\_RndDn = Integer\ of\ Rounded\ down\ PDP\_Max\ to\ the\ nearest\ 1/10^{th}$
    - $PDP\_Max\_RndUp = Integer\ of\ Rounded\ up\ PDP\_Max\ to\ the\ nearest\ 1/10^{th}$
    - $PDP\_Max\_RndDn \leq Port\ Reported\ PDP \leq PDP\_Max\_RndUp$
  - i. In SIDO2, B8...0 (Port Guaranteed PDP) does not match to the PD\_Power\_As\_Source\_Guaranteed VIF field.
  - j. If the Port\_Managed\_Guaranteed\_Type VIF field is set to 1 (Guaranteed) and EPR\_Supported\_As\_SRC is Yes, and B23...16 (Port Maximum PDP) is equal to B15...8 (Port Present PDP) in SIDO1.
  - k. If B23...16 (Port\_Maximum\_PDP) is greater than 60W and B15...8 (Port\_Present\_PDP) is equal to Port\_Maximum\_PDP in SIDO1.

#### TEST.PD.PROT.SRC3.15 *Alert Response Extended Alert.*

Description: As a Sink, the Tester sends *Alert* Message indicating Extended Alert event and verifies that the response messages are correct.

Test Specific Tester Behavior: N/A

Test Conditions:

	Provider Only, P/C, DRP, C/P	Consumer Only
Rev3ChksSrc	✓	

### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1
2. The Tester sends an *Alert* Message to the UUT:
  - a. Type of *Alert* field = Extended Alert Event (Bit 7)
  - b. Extended Alert Event Type = Controller initiated wake e.g Wake on LAN (bits [3..0] = 4)
3. The test stops here when one of the following conditions happens:
  - a. The *Not\_Supported* Message is received as a response to the last *Alert* Message
  - b. The *Get\_Status* Message is received .
4. If No *Get\_Status* Message is received within *tSenderResponse* max after the last *Alert* Message was sent, The Tester sends new *Request* message for 5V, 100 mA to ensure DUT is back to PE\_SRC\_Ready:
  - a. B30...B28 (Object Position) = 001b
  - b. B27 (Reserved) = 0b
  - c. B26 (Capability Mismatch) = 0b
  - d. B25 (USB Communication Capable) = 0b
  - e. B24 (No USB Suspend) = 1b
  - f. B23 (PD3, Unchunked Extended Messages Supported) = 0b
  - g. B22 (EPR Mode Capable) to 0b
  - h. B19...10 (Operating Current) = 100mA
  - i. B9...0 (Maximum Operating Current) = 100mA
5. The check fails if: **[TEST.PD.PROT.SRC3.15#1]**
  - a. The UUT does not respond with an *Accept* or *Wait* Message and the test stops here upon failure.

## 5.4 Test Procedures for Sink Capable UUT

### 5.4.1 PD2 and PD3 Modes

The tests in this subsection are run firstly in the PD2 mode and secondly in the PD3 mode. The tests in this subsection are applicable to UUT with VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider and DRP.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (i.e., a Provider/Consumer or Consumer/Provider) The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- In the PD2 mode:
  - the Tester rejects *VCONN\_Swap*.
- In the PD3 mode:
  - if the Tester is a V<sub>CONN</sub> Source, the Tester accepts the VCONN\_Swap Message.
  - else, the Tester rejects the *VCONN\_Swap* Message.
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### *TEST.PD.PROT.SNK.1 Get\_Sink\_Cap Response*

Description: The Tester verifies that the Sink UUT responds correctly to the *Get\_Sink\_Cap* Message.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only, DRP, P/C, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *Get\_Sink\_Cap* message to the UUT. The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode.
3. The check fails if the UUT does not send *Sink\_Capabilities* Message.  
[TEST.PD.PROT.SNK.1#1]

#### *TEST.PD.PROT.SNK.2 Get\_Source\_Cap Response*

Description: The Tester verifies that the Sink UUT responds correctly to the *Get\_Source\_Cap* Message.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only, P/C, C/P, DRP	Provider Only



<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
- The Tester sends a *Get\_Source\_Cap* Message to the UUT. The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode.
- The Tester checks the response. **[TEST.PD.PROT.SNK.2#1]**
  - The check fails if VIF field PD\_Port\_Type = Consumer Only and the UUT does not send a *Reject* Message (in PD2 mode) or *Not\_Supported* (in PD3 mode).
  - The check fails if VIF field PD\_Port\_Type != Consumer Only and the UUT does not send a *Source\_Capabilities* Message

#### *TEST.PD.PROT.SNK.3 SinkWaitCapTimer Deadline*

Description: The Tester verifies that the UUT provides a Request to a *Source\_Capabilities* Message sent at the deadline limit of *tTypeCSinkWaitCap* after a *Hard Reset*.

Test Specific Tester Behavior: N/A

#### Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
- The Tester sends a *Hard Reset*. It drives V<sub>BUS</sub> to *vSafe0V*, sets Rp default, and then restores V<sub>BUS</sub> to *vSafe5V*.
- The Tester transmits *Source\_Capabilities* Message with single PDO immediately prior to *tTypeCSinkWaitCap* min (the delay is from the time V<sub>BUS</sub> present *vSafe5V* min to the last bit of *Source\_Capabilities* Message EOP):
  - B31...30 (Fixed Supply) set to 00b.
  - B29 (Dual-Role Power) set to 0b, Tester is a Source only.
  - B28 (USB Suspend Supported) set to 0b.
  - B26 (USB Communications Capable) set to 0b.
  - B25 (Dual-Role Data) set to 0b.
  - B24...22 (Reserved) set to 00b.



- g. B21...20 (Peak Current) set to 00b.
- h. B19...10 (Voltage) set to 5V.
- i. B9...0 (Maximum Current) set to 500mA.
- 4. The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode. The Tester checks the response.
  - a. The check fails if the UUT sends a Hard Reset. [TEST.PD.PROT.SNK.3#1]
  - b. If a *Request* Message is received, the test passes and the test stops here.
  - c. If no *Request* Message is received, the test proceeds to the next step.
- 5. The Tester periodically sends *Source\_Capabilities* with *tTypeCSendSourceCap* min interval until *tTypeCSinkWaitCap* max is reached or a *GoodCRC* response is received.
- 6. The check fails if the UUT does not respond with *Request* Message. [TEST.PD.PROT.SNK.3#2]

#### TEST.PD.PROT.SNK.4 SinkWaitCapTimer Timeout

**Description:** The Tester does not send *Source\_Capabilities* Message after cycling V<sub>BUS</sub> to force a *SinkWaitCapTimer* timeout on the UUT, then verifies it is correctly implemented.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, P/C, C/P, DRP	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *Hard Reset*. It drives V<sub>BUS</sub> to *vSafe0V* and then restores V<sub>BUS</sub> to *vSafe5V*.
3. The Tester does not send a *Source\_Capabilities* Message after cycling the V<sub>BUS</sub> to force a *SinkWaitCapTimer* timeout on the UUT.
4. The check issues a warning if the UUT does not send a *Hard Reset* between *tTypeCSinkWaitCap* min and max. [TEST.PD.PROT.SNK.4#1] The delay is between the V<sub>BUS</sub> present *vSafe5V* min and the time of the first bit of Preamble of the *Hard Reset* sent by the UUT.

#### TEST.PD.PROT.SNK.5 SenderResponseTimer Deadline

**Description:** The Tester verifies that the UUT accepts an *Accept* Message sent at the deadline limit of *tSenderResponse* min.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester transmits *Source\_Capabilities* Message with single PDO:
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 0b, Tester is a Source only.
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B26 (USB Communications Capable) set to 0b.
  - e. B25 (Dual-Role Data) set to 0b.
  - f. B24...22 (Reserved) set to 00b.
  - g. B21...20 (Peak Current) set to 00b.
  - h. B19...10 (Voltage) set to 5V.
  - i. B9...0 (Maximum Current) set to 500mA.
3. The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode. The check fails if the UUT does not respond with a *Request* Message.
4. The Tester sends an *Accept* Message at *tCtsSrcAccept* (see **Table 19** Timing Table & Calculations) after the *GoodCRC* Message. This delay is measured from the time the last bit of the *GoodCRC* Message has been transmitted to the time the last bit of the EOP of *GoodCRC* of the *Accept* Message has been received.
5. The check fails if a *Hard Reset* is detected within *tCtsSenderResponse* (see **Table 19** Timing Table & Calculations) after the *GoodCRC* Message sent in respond to the *Request* Message. [TEST.PD.PROT.SNK.5#1]

#### *TEST.PD.PROT.SNK.6 SenderResponseTimer Timeout*

Description: The Tester does not respond to the *Request* Message from the UUT, in order to force a SenderResponseTimer timeout on the UUT and verifies it is correctly implemented.

Test Specific Tester Behavior: N/A

#### Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester transmits *Source\_Capabilities* Message with single.
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 0b, Tester is a Source only.
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B26 (USB Communications Capable) set to 0b.
  - e. B25 (Dual-Role Data) set to 0b.
  - f. B24...22 (Reserved) set to 00b.
  - g. B21...20 (Peak Current) set to 00b.
  - h. B19...10 (Voltage) set to 5V.
  - i. B9...0 (Maximum Current) set to 500mA.
3. The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode. The check fails if a *Request* Message is not received from the UUT.
4. The Tester does not send an *Accept* (as a response to the *Request* Message) in order to force a SenderResponseTimer timeout on the UUT.
5. The Tester checks that a *Hard Reset* is completed between *tSenderResponse* min and *tSenderResponse* max + *tSendHardReset* (see **Table 19** Timing Table & Calculations), the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the last bit of *Hard Reset* EOP has been received. [TEST.PD.PROT.SNK.6#1]

#### *TEST.PD.PROT.SNK.7 PSTransitionTimer Timeout*

Description: The Tester does not send the *PS\_RDY* Message after the *Accept* Message is sent to the UUT, in order to force a PSTransitionTimer timeout on the UUT and verifies it is correctly implemented.

Test Specific Tester Behavior: N/A

#### Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester transmits *Source\_Capabilities* Message with single PDO (5V @ 500mA, Source only).

3. The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode. The check fails if a *Request* Message is not received from the UUT. [TEST.PD.PROT.SNK.7#1]
4. The Tester responds with an *Accept* Message to the *Request* Message.
5. The Tester does not send a *PS\_RDY* Message.
6. The check fails if a *Hard Reset* is not detected within *tPSttransition* min and *tPSttransition* max. [TEST.PD.PROT.SNK.7#2] The delay is between the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message and the first Preamble bit of the *Hard Reset* sent by the UUT.

#### TEST.PD.PROT.SNK.8 Atomic Message Sequence – Accept

**Description:** The Tester sends a *Get\_Sink\_Cap* Message in place of *Accept* Message and verifies the UUT will send a *Soft Reset* and recover from the error.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *Hard Reset*. It drives V<sub>BUS</sub> to *vSafe0V*, sets Rp default, and then restores V<sub>BUS</sub> to *vSafe5V*.
3. The Tester transmits *Source\_Capabilities* Message with single PDO immediately prior to *tTypeCSinkWaitCap* min:
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 0b, Tester is a Source only.
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B26 (USB Communications Capable) set to 0b.
  - e. B25 (Dual-Role Data) set to 0b.
  - f. B24...22 (Reserved) set to 00b.
  - g. B21...20 (Peak Current) set to 00b.
  - h. B19...10 (Voltage) set to 5V.
  - i. B9...0 (Maximum Current) set to 500mA.
4. The check fails if the UUT does not respond with a *Request* Message. [TEST.PD.PROT.SNK.8#1]
5. The Tester sends a *Get\_Sink\_Cap* message at the limit of *tReceiverResponse* max after the reception of *Request* Message. The delay is from the time the last bit of the EOP of the *GoodCRC* corresponding to the *Request* Message.

6. The check fails if a *Soft Reset* Message is not received within *tProtErrSoftReset* max.  
[TEST.PD.PROT.SNK.8#2]
7. The Tester responds with an *Accept* Message to the *Soft Reset* Message.
8. The Tester sends *Source\_Capabilities* Message to the UUT repeatedly until *nCapsCount* is reached or a *GoodCRC* is received. The check fails if *nCapsCount* is reached.
9. The check fails if the UUT does not respond with a *Request* Message.  
[TEST.PD.PROT.SNK.8#3]

#### TEST.PD.PROT.SNK.9 Atomic Message Sequence – PS\_RDY

Description: The Tester sends a *Get\_Sink\_Cap* Message in place of *PS\_RDY* Message and verifies the UUT will send a *Hard Reset*.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *Hard Reset*.
3. The Tester drives V<sub>BUS</sub> to *vSafe0V*, sets Rp to default, and then restores V<sub>BUS</sub> to *vSafe5V*. The Tester transmits *Source\_Capabilities* Message with single PDO (5V @ 500mA, Source only) immediately prior to *tTypeCSinkWaitCap* min.
4. The Tester continues to present Rp default while waiting for a response. The check fails if a *Request* Message is not received from the UUT. [TEST.PD.PROT.SNK.9#1]
5. The Tester responds with an *Accept* Message to the *Request* Message.
6. The Tester sends a *Get\_Sink\_Cap* Message at *tSrcTransReq* max after the reception of the last bit of GoodCRC Message EOP corresponding to the *Accept* Message.
7. The check fails if a *Hard Reset* is not received within *tCtsHardReset* after the last bit of the EOP of the *GoodCRC* Message received. [TEST.PD.PROT.SNK.9#2]

#### TEST.PD.PROT.SNK.10 DR\_Swap Request

Description: As a Sink, the Tester sends a *DR\_Swap* Message, and verifies that the UUT responds correctly.

Test Specific Tester Behavior:

- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message

Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester requests a *DR\_Swap*. The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode.
3. The Tester checks the response. [TEST.PD.PROT.SNK.10#1]
  - a. If the UUT responds with a *Wait* Message, the Tester resends *DR\_Swap* every  $tDRSwapWait + tCtsWait$  for  $nCtsWaitCount$  times. The check fails if the Tester has resent *DR\_Swap* for  $nCtsWaitCount$  times and the test ends here.
  - b. The check fails if
    - i. In PD2, the UUT responds with a *Reject* Message and the VIF field DR\_Swap\_To\_DFP\_Supported = Yes.
    - ii. In PD3, the UUT responds with either.
      1. *Not\_Supported* and (the VIF field DR\_Swap\_To\_DFP\_Supported = Yes or the VIF field DR\_Swap\_To\_UFP\_Supported = Yes) or
      2. *Reject* and the VIF field DR\_Swap\_To\_DFP\_Supported = Yes
 If the check fails, the test ends here.
  - c. The check fails if the UUT responds with an *Accept* Message and the VIF field DR\_Swap\_To\_DFP\_Supported = No.
4. If VIF field DR\_Swap\_To\_DFP\_Supported=N, the test ends here. Otherwise, the Tester requests a further *DR\_Swap*.
5. The Tester checks the response. [TEST.PD.PROT.SNK.10#2]
  - a. If the UUT responds with a *Wait* Message, the Tester resends *DR\_Swap* every  $tDRSwapWait + tCtsWait$  for  $nCtsWaitCount$  times. The check fails if the Tester has resent *DR\_Swap* for  $nCtsWaitCount$  times.
  - b. The check fails if
    - i. In PD2, the UUT responds with a *Reject* Message and the VIF field DR\_Swap\_To\_UFP\_Supported = Yes.
    - ii. In PD3, the UUT responds with either:
      1. *Not\_Supported* and (the VIF fields DR\_Swap\_To\_UFP\_Supported = Yes or the VIF fields DR\_Swap\_To\_DFP\_Supported = Yes) or
      2. *Reject* and the VIF fields DR\_Swap\_To\_UFP\_Supported = Yes
  - c. The UUT responds with an *Accept* Message and the VIF field DR\_Swap\_To\_UFP\_Supported = No.



#### TEST.PD.PROT.SNK.11 VCONN\_Swap Request

**Description:** As a Source, the Tester sends a *VCONN\_Swap* message, and verifies that the UUT responds correctly.

**TEST Specific Tester Behavior:** In the PD3 mode: Retain *SinkTxNG* and send *VCONN\_Swap* within *tSrcHoldBus* after *PS\_RDY* and send the seconds *VCONN\_Swap* Message within *tSrcHoldBus* after the *PS\_RDY* of the earlier *VCONN\_Swap* Message

**Test Conditions:**

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

**Test Procedures:**

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
- The Tester presents Ra to the non-CC line (of the UUT). The Tester checks that VCONN is not present at the non-CC line. [TEST.PD.PROT.SNK.11#1]
- The Tester requests a *VCONN\_Swap*. The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode.
- The Tester checks the response. [TEST.PD.PROT.SNK.11#2]
  - If the UUT responds with a *Wait* Message and the Tester resends *VCONN\_Swap* every *tVCONNSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *VCONN\_Swap* for *nCtsWaitCount* times.
  - The check fails and the test ends here if the UUT responds with a *Reject* Message or *Not\_Supported*, and the VIF field VCONN\_Swap\_To\_On\_Supported = Yes
  - The check fails if the UUT responds with an *Accept* Message and the VIF field VCONN\_Swap\_To\_On\_Supported = No
  - The test ends here if the UUT does not send an *Accept* Message.
- The Tester checks that the UUT sends a *PS\_RDY* Message. [TEST.PD.PROT.SNK.11#3]
- The Tester checks that VCONN is present at the non-CC line. [TEST.PD.PROT.SNK.11#4]
- The Tester requests a further *VCONN\_Swap*.
- The Tester checks the response. [TEST.PD.PROT.SNK.11#5]
  - If the UUT's response is a Wait, the Tester waits *tVCONNSwapWait* + *tCtsWait* seconds to resend *VCONN\_Swap*. The Tester sends *VCONN\_Swap* up to *nCtsWaitCount* times, and the test ends here if the UUT does not respond with an *Accept* or *Reject* Message.
    - In PD3 mode and the UUT responds with a *Reject* Message, the check fails.
    - In PD2 mode and the UUT responds with a *Reject* Message, the check fails if VCONN\_Swap\_To\_Off\_Supported = YES.



9. The Tester sends a *PS\_RDY* Message.
10. The Tester checks that the UUT does not present VCONN at the non-CC line within *tVCONNSourceOff*. [TEST.PD.PROT.SNK.11#6]

#### TEST.PD.PROT.SNK.12 *PR\_Swap* – *PSSourceOffTimer Timeout*

**Description:** As a new Sink, the Tester intentionally does not send a *PS\_RDY* Message after a *PR\_Swap* in order to force a *PSSourceOffTimer* timeout on the UUT and verifies it is correctly implemented.

#### Test Specific Tester Behavior:

- The Tester is a Dual-Role Power, initially a Source.
- The Tester sends *Sink\_Capabilities* setting B27 (Unconstrained Power) to 0b COMMON.PROC.PD.3

#### Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field *PD\_Port\_Type* set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *PR\_Swap* Message. The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode.
3. The Tester checks the response. [TEST.PD.PROT.SNK.12#1]
  - a. If the UUT responds with a *Wait* Message, the Tester resends *PR\_Swap* every *tPRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *PR\_Swap* for *nCtsWaitCount* times.
  - b. If the VIF field *Accepts\_PR\_Swap\_As\_Snk* = No, the check fails if the UUT responds with an *Accept* Message.
  - c. If the VIF field *Accepts\_PR\_Swap\_As\_Snk* = Yes, the check fails if the UUT responds with a *Reject* Message or Not Supported.
  - d. If the VIF field *PD\_Port\_Type* is set to “Consumer Only”, the Tester checks that the UUT responds with a *Reject* Message or Not Supported. The test ends here.
4. At *tSrcTransition* max (this delay is from the last bit of *GoodCRC* sent in response to the *Accept*), the Tester drives the *V<sub>BUS</sub>* voltage to *vSafe0V* within *tSrcSwapStdby* max.
5. The Tester continues to assert *R<sub>p</sub>* (i.e., the Tester does not assert *R<sub>d</sub>* so that Tester can detect the UUT entering Error Recovery). The Tester does not send a *PS\_RDY* Message, in order to force a *PSSourceOffTimer* timeout on the UUT.
6. A USB Type-C Error Recovery is expected from the UUT. The Tester checks the response: [TEST.PD.PROT.SNK.12#2]

- a. The check fails if a USB Type-C Error Recovery is not detected after *tPSSourceOff* max after the time the last bit of the EOP of the *GoodCRC* Message corresponding to the received *Accept* Message.
- b. The check fails if a USB Type-C Error Recovery is detected within *tPSSourceOff* min after the last bit of the EOP of the *GoodCRC* Message corresponding to the received *Accept* Message.

#### *TEST.PD.PROT.SNK.13 PR\_Swap – Request SenderResponseTimer Timeout*

**Description:** As a new Sink, the Tester intentionally does not send a *Request* Message after a *PR\_Swap* in order to force a SenderResponseTimer timeout on the UUT and verifies it is correctly implemented.

#### **Test Specific Tester Behavior:**

- The Tester is a Dual-Role Power, initially a Source
- The Tester sends *Sink Capabilities* setting B27 (Unconstrained Power) to 0b COMMON.PROC.PD.3

#### **Test Conditions:**

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### **Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *PR\_Swap* Message. The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode.
3. The Tester checks the response. [TEST.PD.PROT.SNK.13#1]
  - a. If the UUT responds with a *Wait* Message, the Tester resends *PR\_Swap* every *tPRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *PR\_Swap* for *nCtsWaitCount* times.
  - b. If the VIF field Accepts\_PR\_Swap\_As\_Snk = No, the check fails if the UUT responds with an *Accept* Message.
  - c. If the VIF field Accepts\_PR\_Swap\_As\_Snk = Yes, the check fails if the UUT responds with a *Reject* Message or Not Supported.
  - d. If the VIF field PD\_Port\_Type is set to “Consumer Only”, the Tester checks that the UUT responds with a *Reject* Message or *Not\_Supported* Message. The test ends here.
4. At *tSrcTransition* max (this delay is from the last bit of *GoodCRC* sent in response to the *Accept* Message), the Tester drives the V<sub>BUS</sub> voltage to *vSafe0V* within *tSrcSwapStdby* max.

5. The Tester presents Rd on the CC wire. The Tester sends a *PS\_RDY* at the deadline limit of *tPSSourceOff* min, the delay is from the time the last bit of the EOP of the *GoodCRC* corresponding to the *Accept* Message.
6. The Tester checks that the UUT sends *PS\_RDY* only after the UUT has applied *vSafe5V*. [TEST.PD.PROT.SNK.13#2]
7. The check fails if the *PS\_RDY* from the UUT is not sent within *tPSSourceOn* min. [TEST.PD.PROT.SNK.13#3] This delay is from the *GoodCRC* sent by the UUT in response to the *PS\_RDY* sent by the Tester.
8. The Tester checks that the UUT sends a *Source\_Capabilities* Message after *tSwapSourceStart* min. [TEST.PD.PROT.SNK.13#4]
9. Once the Tester receives *Source\_Capabilities* Message, the Tester establishes a PD contract using common procedure COMMON.PROC.PD.11.
10. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT.
11. The Tester sends a *Get\_Source\_Cap* Message to the UUT.
12. The check fails if the Tester does not receive *Source\_Capabilities* Message. [TEST.PD.PROT.SNK.13#5]
13. The Tester intentionally does not send a *Request* Message and waits for a *Hard Reset* Signaling.
14. The Tester checks that a *Hard Reset* Signaling is completed between *tSenderResponse* min and *tSenderResponse* max + *tSendHardReset* (see **Table 19** Timing Table & Calculations) the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the last bit of *Hard Reset* EOP has been received. [TEST.PD.PROT.SNK.13#6]

#### TEST.PD.PROT.SNK.14 Valid Use of GoodCRC on Power up

Description: The Tester sends *Source\_Capabilities* at the earliest opportunity and checks that the UUT does not send *GoodCRC* before it is completely ready for operation.

Test Specific Tester Behavior: N/A

#### Test Conditions:

	Consumer Only, DRP, C/P	Provider Only, P/C
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. The test starts in a disconnected state.
2. The tester applies Rp default.
3. If the UUT attachment is detected, the Tester applies VCONN (if Ra is detected) and *vSafe5V* on V<sub>BUS</sub>.
4. The Tester waits for 1 ms.
5. The Tester transmits *Source\_Capabilities* Message with single PDO:
  - a. B31...30 (Fixed Supply) set to 00b.

- b. B29 (Dual-Role Power) set to 0b.
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B27 (Unconstrained Power) set to 1b.
  - e. B26 (USB Communications Capable) set to 0b.
  - f. B25 (Dual-Role Data) set to 0b.
  - g. B24 (PD3, Unchunked Extended Messages Supported) set to 0b.
  - h. B23 (EPR Mode Capable) to 0b, unless it is mentioned in the test procedure.
  - i. B21...20 (Peak Current) set to 00b.
  - j. B19...10 (Voltage) set to 5V.
  - k. B9...0 (Maximum Current) set to 500mA.
6. Repeat Step-5 if the Tester does not receive a *GoodCRC* from the UUT in response to *Source\_Capabilities* Message. This requires a delay of *tTypeC\_SendSourceCap* (min to max). This sequence is repeated at least 50 times.
  7. The check fails if the UUT does not respond with a *Request* Message, within *tReceiverResponse* of the UUT sending the first *GoodCRC*. [TEST.PD.PROT.SNK.14#1]
  8. The Tester responds with an *Accept* Message to the *Request* Message.
  9. When the V<sub>BUS</sub> voltage is stable within the target voltage, the Tester sends a *PS\_RDY* Message at *tSrcTransReq* max after the reception of the last bit *GoodCRC* Message EOP corresponding to the *Accept* Message.
  10. An explicit contract is now established.

#### 5.4.2 PD3 Mode

The tests in this subsection are run only in PD3 mode.

The tests in this subsection are applicable to UUT with VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider and DRP.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (ie, a Provider/Consumer or Consumer/Provider). The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### TEST.PD.PROT.SNK3.1 Get\_Source\_Cap\_Extended

Description: The Tester verifies that the Sink UUT responds correctly to *Get\_Source\_Cap\_Extended* Message.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev3ChkdSnk</i>	✓	
<i>Rev3UnchkdSnk</i>	✓	

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *Get\_Source\_Cap\_Extended* Message to the UUT and continues to present *SinkTxNG* while waiting for a response.
3. The check fails if the UUT does not send either *Not\_Supported* or *Source\_Capabilities\_Extended* Message. [TEST.PD.PROT.SNK3.1#1]

*TEST.PD.PROT.SNK3.2 Alert Response Source Input Change*

Description: As a Source, the Tester sends *Alert* Message indicating OTP event and verifies that the response messages are correct.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev3ChkdSnk</i>	✓	

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends an *Alert* Message to the UUT with Type of *Alert* field = OTP event (Bit 3).
3. The Tester presents *SinkTxOK*.
4. The test passes if one of the following conditions happens [TEST.PD.PROT.SNK3.2#1]
  - a. A *Not\_Supported* Message is received as a response to the last *Alert* Message and the UUT as a Sink has no sink PPS APDO in VIF.
  - b. A *Get\_Status* Message is received within *tSenderResponse* min as a response to the last *Alert* Message.
  - c. No message is received for *tSenderResponse* min
 Otherwise, the test fails and stops here.

**NOTE:** *Alert Shall* be supported when *SPR PPS Mode* is supported

	Not_supported	Get_status	No message
Sink APDO in VIF	Fail	pass	pass
No APDO in VIF	Pass	Pass	pass

#### TEST.PD.PROT.SNK3.3 Alert Response Battery Status Change

**Description:** As a Source, the Tester sends *Alert* Message indicating Battery Status Change and verifies that the response messages are correct.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev3ChkdSnk</i>	✓	
<i>Rev3UnchkdSnk</i>	✓	

**Test Procedures:**

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
- The Tester sends an *Alert* Message to the UUT
  - Type of *Alert* field = Battery Status Change Event (Bit 1)
  - Fixed Batteries = Battery 0 (Bit20)
- The Tester presents *SinkTxOK*.
- The test passes if one of the following conditions happens [TEST.PD.PROT.SNK3.3#1]:
  - A *Not\_Supported* Message is received as a response to the last *Alert* Message and the UUT as a Sink does not have sink PPS APDO in VIF.
  - A *Get\_Battery\_Status* Message is received within *tSenderResponse* min after the Alert Message was sent.
  - No message is received for *tSenderResponse* min.
 Otherwise, the test fails and stops here.

**NOTE:** *Alert* Shall be supported when SPR PPS Mode is supported.

	Not_supported	Get_Battery_status	No message
Sink APDO in VIF	Fail	Pass	Pass
No APDO in VIF	Pass	Pass	Pass

#### TEST.PD.PROT.SNK3.4 Soft Reset Sent Regardless of Rp Value

**Description:** The Tester verifies that the Sink UUT responds correctly to a missing *GoodCRC*.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *Get\_Sink\_Cap* Message to the UUT and continues to present *SinkTxNG* while waiting for a response.
3. Upon receipt of the *Sink\_Capabilities* Message, the Tester doesn't reply with *GoodCRC*.
4. The check fails if a *Soft Reset* Message is not sent by the UUT within  $t_{Receive} \text{ max} + t_{SoftReset} \text{ max}$ . [TEST.PD.PROT.SNK3.4#1] The delay is measured from the time the last bit of the last retransmitted *Sink\_Capabilities* Message EOP has been received to the time the last bit of the *Soft Reset* Message EOP has been received.

#### TEST.PD.PROT.SNK3.5 Sink PPS Normal Operation

Description: The Tester verifies that the Sink UUT verifies the UUT periodically re-request the PPS APDO.

Test Specific Tester Behavior: N/A

#### Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.12
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.5
2. The Tester presents *SinkTxOk*.
3. The check fails if a *Request* Message is not received within  $t_{PPSRequest} \text{ max}$ . [TEST.PD.PROT.SNK3.5#1] The delay is measured from the time the last bit of the EOP of the *GoodCRC* Message sent by the Tester in response to the previous *Request* Message to the time the first bit of the Preamble of the new *Request* Message is received.
4. The Tester runs the following additional field checks on the Request Data Object: [TEST.PD.PROT.SNK3.5#2]
  - a. Object Position field is 2.



5. The Tester intentionally sends a *Wait* Message. The test fails if another *Request* Message is received within min of *tSinkRequest*. This is measured when the EOP of a *Wait* Message has been received. [TEST.PD.PROT.SNK3.5#3]
6. The Tester shall continue monitoring for *tPPSTimeout* max that the UUT is not disconnecting. Disconnect is deemed to have taken place when the CC line voltage falls outside the 'connect' range, because of Rd being released by the UUT. If the UUT sends another *Request* Message during this time, the Tester will respond with a *Wait* Message. [TEST.PD.PROT.SNK3.5#4]
7. The check fails if a *Request* Message is not received within *tPPSRequest* max. The delay is measured from the last bit of the *EOP* of the *GoodCRC* Message sent by the Tester in response to the previous *Request* Message. [TEST.PD.PROT.SNK3.5#5]

#### TEST.PD.PROT.SNK3.6 Revision Number Test

Description: As a Source, the Tester sends *Source\_Capabilities* Message with the Message Header containing a higher revision number than that supported and the UUT responds using the highest supported revision number.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only, C/P, DRP	Provider Only, P/C
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. The test starts in a disconnected state.
2. The tester applies Rp default.
3. If the UUT attachment is detected, the Tester applies VCONN (if Ra is detected) and *vSafe5V* on V<sub>BUS</sub>.
4. The Tester waits for 50 ms.
5. The Tester sets the Specification Revision (in the Message Header) to 11b and transmits *Source\_Capabilities* Message with single PDO:
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 0b.
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B27 (Unconstrained Power) set to 1b.
  - e. B26 (USB Communications Capable) set to 0b.
  - f. B25 (Dual-Role Data) set to 0b.
  - g. B24...22 (Reserved) set to 00b.
  - h. B21...20 (Peak Current) set to 00b.
  - i. B19...10 (Voltage) set to 5V.
  - j. B9...0 (Maximum Current) set to 500mA.
6. Repeat Step-5 if the Tester does not receive a *GoodCRC* from the UUT in response to *Source\_Capabilities* Message. This requires a delay of *tTypeC\_SendSourceCap* (min to max). This sequence is repeated at least nCapsCount times.

- The check fails if a *Request* Message is not received from the UUT.  
[TEST.PD.PROT.SNK3.6#1]

#### TEST.PD.PROT.SNK3.7 GoodCRC Specification Revision Compatibility

**Description:** As a Source, the Tester verifies the UUT accepts *GoodCRC* with Specification Revision set to 00b, 01b, or 10b.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, C/P, DRP	Provider Only, P/C
<i>Rev3ChkdSnk</i>	✓	

**Test Procedures:**

- The test starts in a disconnected state.
- The tester applies Rp default.
- If the UUT attachment is detected, the Tester applies VCONN (if Ra is detected) and *vSafe5V* on V<sub>BUS</sub>.
- The Tester waits for 50 ms.
- The Tester transmits *Source\_Capabilities* Message with single PDO:
  - B31...30 (Fixed Supply) set to 00b.
  - B29 (Dual-Role Power) set to 0b.
  - B28 (USB Suspend Supported) set to 0b.
  - B27 (Unconstrained Power) set to 1b.
  - B26 (USB Communications Capable) set to 0b.
  - B25 (Dual-Role Data) set to 0b.
  - B24...22 (Reserved) set to 00b.
  - B21...20 (Peak Current) set to 00b.
  - B19...10 (Voltage) set to 5V.
  - B9...0 (Maximum Current) set to 500mA.
- Repeat Step-5 if the Tester does not receive a *GoodCRC* from the UUT in response to *Source\_Capabilities* Message. This requires a delay of *tTypeC\_SendSourceCap* (min to max). This sequence is repeated at least *nCapsCount* times.
- The check fails if the UUT does not respond with a *Request* Message.  
[TEST.PD.PROT.SNK3.7#1]
- The Tester sends a *GoodCRC* (acknowledging the *Request* Message) with Specification Revision set to 00b. The check fails if the UUT retransmits the *Request* Message.  
[TEST.PD.PROT.SNK3.7#2]
- Repeat the test with Specification Revision in *GoodCRC* set to 01b and 10b.

#### TEST.PD.PROT.SNK3.8 the GotoMin Message

This test has been deprecated

#### TEST.PD.PROT.SNK3.9 Alert Response Extended Alert

**Description:** As a Source, the Tester sends the *Alert* Message indicating Extended Alert event and verifies that the response messages are correct.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only, P/C, DRP, C/P	Provider Only
Rev3ChksSnk	✓	

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends an *Alert* Message to the UUT:
  - a. Type of *Alert* field = Extended Alert Event (Bit 7)
  - b. Extended Alert Event Type = Power state change (bits [3..0] = 1)
3. The Tester presents *SinkTxOK* after *tSrcHoldsBus*.
4. The test stops here when one of the following conditions happens:
  - a. The *Not\_Supported* Message is received as a response to the last *Alert* Message
  - b. The *Get\_Status* Message is received.
5. If the *Get\_Status* Message has not been received within *tSenderResponse* max after the last *Alert* Message was sent, The Tester sends new *Source\_Capabilities* Message with a single PDO to ensure DUT is back to PE\_SNK\_Ready:
  1. B31...30 (Fixed Supply) set to 00b.
  2. B29 (Dual-Role Power) set to 1b.
  3. B28 (USB Suspend Supported) set to 0b.
  4. B27 (Unconstrained Power) set to 1b.
  5. B26 (USB Communications Capable) set to 0b.
  6. B25 (Dual-Role Data) set to 0b.
  7. B24 (PD3, Unchunked Extended Messages Supported) set to 0b.
  8. B23 (EPR Mode Capable) to 0b, unless it is mentioned in the test procedure.
  9. B21...20 (Peak Current) set to 00b.
  10. B19...10 (Voltage) set to 5V.
  11. B9...0 (Maximum Current) set to 500mA.
6. The check fails if the UUT does not respond with the *Request* Message [TEST.PD.PROT.SNK3.9#1] and the test stops here upon failure.

## 5.5 Structured VDM Test Procedures for Source Capable UUT

### 5.5.1 PD2 and PD3 Modes

The tests in this subsection are run firstly in PD2 mode and secondly in PD3 mode.

The tests in this subsection are applicable to UUT with VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer and DRP.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (ie, a Provider/Consumer or Consumer/Provider). The Tester always replies with a *Reject* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### TEST.PD.VDM.SRC.1 Discovery Process and Enter Mode

Description: The Tester verifies that the UUT responds appropriately to VDM messages.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only	Provider Only, C/P, P/C, DRP
<i>Rev2Src</i>		✓
<i>Rev3ChkdSrc</i>		✓

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester sends *Request* Message with B25 (USB Communication Capable) set to 1b.
3. The Tester sends a *Discover Identity* Command to the UUT, using SOP. The Tester checks the response:
  - a. In PD2 mode, the Tester checks the response as follows:  
[TEST.PD.VDM.SRC.1#1]
    - i. The UUT Supports Structured VDM is NO: the check passes if the UUT Ignores the Message within *tVDMSenderResponse* (i.e. if UUT replies with either NAK, ACK, BUSY, Reject, or any AMS will cause check to fail).
    - ii. The UUT Supports Structured VDM is YES: the check fails if the UUT is DFP, and the UUT replies with anything but NAK within *tVDMSenderResponse*.
  - b. In PD3 mode, the Tester checks the response as follows:  
[TEST.PD.VDM.SRC.1#2]

- i. The check fails if the UUT responds with anything but a *Discover Identity* Response message or *Not\_Supported*, and the Tester has resent *Discover Identity* Command *nDiscoverIdentityCount* min after it is in PE\_SRC\_RDY for *nBusyCount* times.
    - ii. If Supports Structured VDMs is Yes, the check fails if the UUT responds with a “Responder BUSY” message, and the Tester has resent *Discover Identity* Command for *nBusyCount* times every *tVDMBusy* min.
    - iii. The check fails if Supports Structured VDMs is YES and the UUT responds with Not Supported.
    - iv. The check fails if Supports Structured VDMs is NO and the UUT responds with “Responder ACK”, “Responder NAK” or “Responder BUSY”.
    - v. The check fails if Responds\_To\_Discov\_SOP\_DFP is NO and the UUT responds with “Responder ACK”.
    - vi. The check fails if Responds\_To\_Discov\_SOP\_DFP is YES and the UUT responds with “Responder NAK”.
  - c. If the UUT sends a “Responder ACK” message, the Tester performs the following check on the “Responder ACK” message: **[TEST.PD.VDM.SRC.1#3]**
    - i. ID Header VDO:
      - 1. The check fails if B31 does not match VIF field Data\_Capable\_as\_USB\_Host\_SOP
      - 2. The check fails if B30 does not match VIF field Data\_Capable\_as\_USB\_Device\_SOP
      - 3. Check fails if B29..27 if:
        - a. In PD2 Mode:
          - i. Field is not set to Undefined if VIF Product\_Type\_UFP\_SOP is PSD
          - ii. Does not match VIF Product\_Type\_UFP\_SOP otherwise.
        - b. In PD3 Mode:
          - Does not match VIF Product\_Type\_UFP\_SOP
      - 4. The check fails if B26 does not match VIF field Modal\_Operation\_Supported\_SOP
      - 5. In PD3, the check fails if B25...23 does not match VIF field Product\_Type\_DFP\_SOP
      - 6. In PD3, the check fails if B22...21 does not match VIF field ID\_Header\_Connector\_Type
      - 7. In PD3, the check fails if B20...16 is not set to zero.
      - 8. In PD2, the check fails if B25...16 is not set to zero.
      - 9. The check fails if B15...0 does not match VIF field USB\_VID\_SOP
    - ii. The check fails if Cert Stat VDO does not match VIF field XID\_SOP
    - iii. The check fails if Product VDO does not match VIF fields PID\_SOP and bcdDevice\_SOP
- 4. The Tester sends an *Attention* Command to the UUT.

5. The Tester waits for *tCtsSenderResponse* (see **Table 19** Timing Table & Calculations), and the check fails if supports Structured VDMs is Yes and the UUT sends anyResponse message. [TEST.PD.VDM.SRC.1#4]

#### TEST.PD.VDM.SRC.2 Invalid Fields – Discover Identity

**Description:** The Tester verifies that the UUT responds correctly to a *Discover Identity* Command sent with invalid fields.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, C/P, P/C, DRP
<i>Rev2Src</i>		✓
<i>Rev3ChkdSrc</i>		✓

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester instead sends *Request* Message with B25 (USB Communication Capable) set to 1b.
3. The Tester sends *Discover Identity* Command with SVID = 0xEEEE. The Tester checks the response. [TEST.PD.VDM.SRC.2#1]
  - a. If Supports Structured VDMs is NO, then the UUT response shall be Ignore in PD2 and *Not\_Supported* in PD3.
  - b. Otherwise, the check fails if the UUT does not reply with a “Responder NAK”. The check fails if the SVID is not set to 0xEEEE in the “Responder NAK” message.

## 5.6 Structured VDM Test Procedures for Sink Capable UUT

### 5.6.1 PD2 and PD3 Modes

The tests in this subsection are run firstly in PD2 mode and secondly in PD3 mode.

The tests in this subsection are applicable to UUT with VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider and DRP.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (ie, a Provider/Consumer or Consumer/Provider) The Tester always replies with a *Reject* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

### TEST.PD.VDM.SNK.1 Discovery Process and Enter Mode

**Description:** The Tester verifies that the UUT responds correctly to the Discovery Process and the *Enter Mode* Commands.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, P/C, C/P, DRP	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

**Test Procedures:**

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
- The Tester sends a *Discover Identity* Command to the UUT, using SOP. The Tester checks the response. [TEST.PD.VDM.SNK.1#1]
  - The UUT shall respond with a “Responder NAK”, if VIF fields Responds\_To\_Discov\_SOP\_UFP = NO and Supports Structured VDMs is Yes. Test ends here.
  - The UUT shall respond with either *Not Supported* (in PD3 mode) or Ignore (in PD2 mode), if Supports Structured VDMs is NO. Test ends here.
  - If the UUT interrupts the Discovery Process (by sending anything but a *Discover Identity* Response message), the Tester resends *Discover Identity* Command *tVDMBusy* min after it is in PE\_SNK\_RDY for *nBusyCount* times. The check fails and the test ends here if the Tester has resent *Discover Identity* Command for *nBusyCount* times.
  - The check fails if the UUT responds with a “Responder BUSY” message and the Tester has resent *Discover Identity* Command for *nBusyCount* times every *tVDMBusy* min. Test ends here.
  - The check fails if the UUT responds with a “Responder ACK” message and VIF field Responds\_To\_Discov\_SOP\_UFP= No. Test ends here.
  - The check fails if the UUT does not respond with ‘Responder ACK’ message and VIF field Responds\_To\_Discov\_SOP\_UFP=Yes. Test ends here.
- Perform the following check on “Responder ACK” message: [TEST.PD.VDM.SNK.1#2]
  - Number of VDOs:
    - The check fails if the number of VDOs in the message does not match the value stated in the **Table 10**, 11, 12 and 13 below, as appropriate, based on the Product Type(s) found in the message.
    - The check fails if the number of VDOs is 7, and the 6<sup>th</sup> VDO (Pad) is not all 0’s.
  - ID Header VDO:
    - The check fails if B31 does not match VIF field Data\_Capable\_as\_USB\_Host\_SOP



- ii. The check fails if B30 does not match VIF field  
Data\_Capable\_as\_USB\_Device\_SOP
  - iii. The check fails if B29...27 if
    - a. In PD2 Mode:
      - i. Field is not set to Undefined if VIF  
Product\_Type\_UFP\_SOP is PSD
      - ii. Field is set to AMA and VIF  
Product\_Type\_UFP\_SOP is not set to peripheral.
      - iii. Does not match VIF Product\_Type\_UFP\_SOP  
otherwise.
    - b. In PD3 Mode:
      - Does not match VIF Product\_Type\_UFP\_SOP
  - iv. The check fails if B26 does not match VIF field  
Modal\_Operation\_Supported\_SOP
  - v. In PD3 mode, the check fails if B25...23 does not match VIF field  
Product\_Type\_DFP\_SOP
  - vi. In PD3, the check fails if B22...21 does not match VIF field  
ID\_Header\_Connector\_Type
  - vii. In PD3 mode, the check fails if B20...16 is not set to zero.
  - viii. In PD2 mode, the check fails if B25...16 is not set to zero.
  - ix. The check fails if B15...0 does not match VIF field USB\_VID\_SOP
  - c. The check fails if Cert Stat VDO does not match VIF field XID\_SOP
  - d. The check fails if Product VDO does not match VIF fields PID\_SOP and  
bcdDevice\_SOP
- 4. The Tester sends a *Discover SVIDs* Command to the UUT, using SOP. The Tester checks the response. [TEST.PD.VDM.SNK.1#3]
  - a. The check fails if the UUT responds with either *Not\_Supported* or Ignore. The test stops here.
  - b. If the UUT interrupts the Discovery Process (by responding with anything but a *Discover SVIDs* Response message), the Tester resends *Discover SVIDs* Command *tVDMBusy* min after it is in PE\_SNK\_RDY for *nBusyCount* times. The check fails and the test ends here if the Tester has resent *Discover Identity* Command for *nBusyCount* times.
  - c. If the UUT responds a “Responder BUSY” message, the Tester resends *Discover SVIDs* Command for *nBusyCount* times every *tVDMBusy* min. Once the Tester has repeated *Discover SVIDs* for *nBusyCount* times, the check fails, and test stops here.
  - d. If the UUT responds with a “Responder NAK” message, the Tester checks that VIF field Modal\_Operation\_Supported\_SOP = No.
  - e. If the UUT responds with a “Responder ACK” message, the Tester checks that VIF field Modal\_Operation\_Supported\_SOP = Yes.
- 5. The Tester evaluates the number of SVIDs in the *Discover SVIDs* ACK message and checks the following: [TEST.PD.VDM.SNK.1#4]
  - a. Each discovered SVID appears in VIF SOPSVIDList. SVID’s position in the VDO is checked according to SVID\_Fixed\_SOP value in VIF.

- i. If SVID\_Fixed\_SOP is set to YES, the order of SVIDs in the VDO matches the order of SOPSVID elements in VIF SOPSVIDList.
    - ii. If SVID\_Fixed\_SOP is set to NO, each SVID in the VDO list can be in any position in the VIF SOPSVIDList.
  - b. The number of SVIDs discovered lies between Num\_SVIDs\_min\_SOP and Num\_SVIDs\_max\_SOP.
  - c. For each VDO, the Tester checks if B15...0 (SVID n+1) are set to zeros, this marks the last VDO.
  - d. For each VDO, the Tester checks if B31...16 (SVID n) are set to zeros, then B15...0 shall also be zeros, and this marks the last VDO.
  - e. If the *Discover SVIDs* ACK message has 6 VDOs, and B15...0 in VDO 6 is non-zero, then repeat Step-3 until the Tester finds the last VDO.
6. For each SVID:
  - a. The Tester sends a *Discover Modes* Command to the UUT, using SOP.
  - b. The Tester checks the response: [TEST.PD.VDM.SNK.1#5]
    - i. The check fails if the UUT responds with either *Not Supported* or Ignore. The test stops here.
    - ii. If the UUT interrupts the Discovery Process (by responding with anything but a *Discover Modes* Response message), the Tester resends *Discover SVIDs* Command *tVDMBusy* min after it is in PE\_SNK\_RDY for *nBusyCount* times. The check fails and the test ends here if the Tester has resent *Discover Identity* Command for *nBusyCount* times.
    - iii. If the UUT responds a "Responder BUSY" message, the Tester resends *Discover Modes* Command for *nBusyCount* times every *tVDMBusy* min. Once the Tester has repeated *Discover SVIDs* for *nBusyCount* times, the check fails, and test stops here.
    - iv. If the UUT responds with a "Responder NAK" message, the Tester checks that Modal\_Operation\_Supported\_SOP = No.
    - v. If the UUT responds with a "Responder ACK" message, the Tester checks that Modal\_Operation\_Supported\_SOP = Yes.
    - vi. The number of Modes discovered lies between SVID\_Num\_Modes\_Min\_SOP and SVID\_Num\_Modes\_Max\_SOP.
7. For every Mode in each SVID:
  - a. The Tester sends *Enter Mode* to the UUT, using SOP. If the corresponding VIF SOPSVIDMode element contains SVID\_Mode\_VDO\_SOP, the tester shall append the SVID\_Mode\_VDO\_SOP to the VDM.
  - b. The Tester checks the response. [TEST.PD.VDM.SNK.1#6]
    - i. If the UUT responds with "Responder BUSY", the check fails.
    - ii. If the UUT responds with a "Responder NAK", the test continues.
    - iii. If the UUT responds with "Responder ACK" to *Enter Mode*, the Tester then sends an *Exit Mode* Command to the UUT using SOP. The check fails if the UUT does not respond with "Responder ACK" to Exit Mode.
8. The Tester sends an *Attention* Command to the UUT.
9. The Tester waits for *tCtsSenderResponse* (see Table 19 Timing Table & Calculations), and the check fails if supports Structured VDMs is Yes and the UUT sends any response message. [TEST.PD.VDM.SNK.1#7]

Table 10 Rev 2 Port - Number of VDOs Expected

<b>Product Type (UFP)</b>	Undefined	4
	PD USB Hub	4
	PD USB Peripheral	4
	AMA	5

Table 11 Rev 2 Cable - Number of VDOs Expected

<b>Product Type</b>	PassiveCable	5
	Active Cable	5

Table 12 Rev 3 Port Number of VDOs Expected

<b>Product Type (UFP)</b>	<b>Product Type (DFP)</b>				
		Undefined	PD USB Hub	PD USB Host	Power Brick
	Undefined	4	5	5	5
	PD USB Hub	5	7*	7*	7*
	PD USB Peripheral	5	7*	7*	7*
	PSD	4	5	5	5

\*The 6<sup>th</sup> VDO (Pad) must be all 0's

Table 13 Cable - Number of VDOs Expected

<b>Product Type</b>	PassiveCable	5
	Active Cable	6
	VPD	5

#### TEST.PD.VDM.SNK.2 Exit Mode without Entering

**Description:** The Tester verifies that the UUT responds with a correctly formed message to an *Exit Mode* Command from the Tester.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, P/C, C/P, DRP	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

**Test Procedures:**

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
- The Tester sends a *Discover SVIDs* Command to the UUT, using SOP.
- The Tester checks the response: [TEST.PD.VDM.SNK.2#1]

- a. If the UUT responds with either *Not Supported* (in PD3 mode) or Ignore (in PD2 mode), the Tester checks that the UUT does not Support Structured VDMs. The test ends here.
  - b. If the UUT interrupts the Discovery Process (by responding with anything but a *Discover SVIDs* Response message), the Tester resends *Discover SVIDs* Command *tVDMBusy* min after it is in PE\_SNK\_RDY for *nBusyCount* times. The check fails and the test ends here if the Tester has resent *Discover Identity* Command for *nBusyCount* times.
  - c. If the UUT responds with a BUSY message, the Tester resends *Discover SVIDs* Command for *nBusyCount* times every *tVDMBusy* min. Once the Tester has repeated *Discover SVIDs* Command for *nBusyCount* times, the check fails, and the test ends here.
  - d. If the UUT responds with a “Responder NAK” message, the Tester checks that VIF field Modal\_Operation\_Supported\_SOP = No. The test ends here.
  - e. If the UUT responds with a “Responder ACK” message, the Tester checks that VIF field Modal\_Operation\_Supported\_SOP = Yes.
4. The Tester sends *Exit Mode* to the UUT using SOP, set to the first SVID and the first Mode.
  5. The Tester checks that *Exit Mode* NAK as a response is sent before *tVDMExitMode* max. [TEST.PD.VDM.SNK.2#2] This delay is from the last bit of the EOP of the *GoodCRC*.

#### TEST.PD.VDM.SNK.3 Interruption by PD Message

#### TEST.PD.VDM.SNK.4 Interruption by VDM Message

**NOTE:** The following sections have been removed due to changes in the main specifications

#### TEST.PD.VDM.SNK.5 DR Swap in Modal Operation

**Description:** The Tester verifies that the UUT issues a *Hard Reset* within *tDRSwapHardReset* if *DR\_Swap* is sent by during a modal operation.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, P/C, C/P, DRP	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *Discover SVIDs* Command to the UUT, using SOP.
3. The Tester checks the response: [TEST.PD.VDM.SNK.5#1]

- a. If the UUT responds with either *Not\_Supported* (in PD3 mode) or Ignore (in PD2 mode), the Tester checks that the UUT does not Support Structured VDMs. The test ends here.
  - b. If the UUT interrupts the Discovery Process (by responding with anything but a *Discover SVIDs* Response message), the Tester resends *Discover SVIDs* Command *tVDMBusy* min after it is in PE\_SNK\_RDY for *nBusyCount* times. The check fails and the test ends here if the Tester has resent *Discover Identity* Command for *nBusyCount* times.
  - c. If the UUT responds with a BUSY message, the Tester resends *Discover SVIDs* Command for *nBusyCount* times every *tVDMBusy* min. Once the Tester has repeated *Discover SVIDs* Command for *nBusyCount* times, the check fails, and the test ends here.
  - d. If the UUT responds with a “Responder NAK” message, the Tester checks that VIF field Modal\_Operation\_Supported\_SOP = No. The test ends here.
  - e. If the UUT responds with a “Responder ACK” message, the Tester checks that VIF field Modal\_Operation\_Supported\_SOP = Yes.
4. The test ends here if the UUT does not send *Discover SVIDs* ACK message.
  5. For each SVID
    - a. The Tester sends a *Discover Modes* Command to the UUT, using SOP.
    - b. The Tester checks the response: [TEST.PD.VDM.SNK.5#2]
      - i. The check fails if a response message (ACK, NAK or BUSY) is not sent.
      - ii. If the UUT responds with a BUSY message, the Tester resends *Discover Modes* Command for *nBusyCount* times every *tVDMBusy* min. Once the Tester has repeated *Discover Modes* Command for *nBusyCount* times, the check fails.
      - iii. The check fails if the UUT does not responds with a *Discover Modes* ACK message using SOP.
  6. For any first found SOPSVIDMode element in the SOPSVIDModeList where SVID\_Mode\_Enter = Yes in the VIF file:
    - a. The Tester sends *Enter Mode* to the UUT, using SOP. If the VIF SOPSVIDMode element contains SVID\_Mode\_VDO\_SOP, the tester shall append SVID\_Mode\_VDO\_SOP to the VDM.
    - b. The Tester checks the response: [TEST.PD.VDM.SNK.5#3]
      - i. The check fails if a response message (ACK or NAK) is not sent.
      - ii. If the UUT responds with an *Enter Mode* NAK, the test ends here.
    - c. If the UUT responds with an *Enter Mode* ACK, the Tester sends a *DR\_Swap* using SOP packet. The check fails if *Hard Reset* is not received within *tDRSwapHardReset*. [TEST.PD.VDM.SNK.5#4]

#### *TEST.PD.VDM.SNK.6 Structured VDM Revision Number Test*

Description: The Tester verifies that the UUT correctly handles Structured VDM Version higher than supported.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only, P/C, C/P, DRP	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *Discover Identity* Command to the UUT, using SOP. The Structured VDM Version field in the VDM Header is set to 3.
3. The Tester checks the response [TEST.PD.VDM.SNK.6#1]
  - a. If the UUT responds with either *Not\_Supported* (in PD3 mode) or Ignore (in PD2 mode), the Tester checks that Supports\_Structured\_VDMs is NO.
  - b. Otherwise, the Tester checks that a response message (ACK, NAK or BUSY) is sent.

#### *TEST.PD.VDM.SNK.7 Unrecognized VID in Unstructured VDM*

Description: As a DFP, the Tester verifies that the UUT responds a *Not\_Supported* Message to an unstructured VDM message with an unrecognized VID.

Test Specific Tester Behavior: N/A

#### Test Conditions:

	Consumer Only, C/P , P/C, DRP	Provider Only
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends an unstructured VMD message to the UUT, in which VID in the Unstructured VDM Header is set to the Tester manufacture's VID.
3. If the UUT does not respond with either *Not\_Supported* (in PD3 mode) or Ignore (in PD2 mode), the check fails. [TEST.PD.VDM.SNK.7#1]

## 5.7 Structured VDM Test Procedures for Cable Plug UUT

### 5.7.1 PD2 and PD3 Modes

The tests in this subsection are run firstly in PD2 mode and secondly in PD3 mode.

The tests in this subsection are applicable to UUT with VIF fields:

VIF\_Product\_Type = Cable OR  
VIF\_Product\_Type = 0 (Port Product) AND PD\_Port\_Type = 5 (eMarker) AND Product\_Type = 6 (V<sub>CONN</sub> Powered Device).

#### *TEST.PD.VDM.CBL.1 Discovery Process and Enter Mode*

**Description:** The Tester verifies that the UUT responds correctly to the Discovery Process and the *Enter Mode* Commands.

#### **Test Conditions:**

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev2VcMinEnd1</i>				✓	✓
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev2VcMaxEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev2VcMinEnd2</i>				✓	
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev2VcMaxEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

#### **Test Procedures:**

1. There are 4 possible bring-up procedures:
  - a. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - b. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - c. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a *Discover Identity* Command to the UUT, using SOP'.
3. The Tester checks the response message: [TEST.PD.VDM.CBL.1#1]
  - a. The check fails if the UUT responds anything but a *Discover Identity* Response message.



- b. If the UUT responds with “Responder BUSY” message, the Tester resends *Discover Identity* Command for *nBusyCount* times every *tVDMBusy* min. Once the Tester has repeated *Discover Identity* for *nBusyCount* times, the check fails.
- c. If the UUT responds with a “Responder NAK”, the check fails.
- d. Check the “Responder ACK” message:
  - i. In PD2 mode  
The check fails if the Number of Data Objects is not 5.
  - ii. In PD3 mode  
If Product Type (ID Header B29...27) = Passive Cable, the check fails if the Number of Data Objects is not 5.  
If Product Type (ID Header B29...27) = Active Cable, the check fails if the Number of Data Objects is not 6.  
If Product Type (ID Header B29...27) = VPD, the check fails if the Number of Data Objects is not 5.
  - iii. ID Header VDO:
    - 1. The check fails if B31 does not match VIF field `Data_Capable_as_USB_Host`
    - 2. The check fails if B30 does not match VIF field `Data_Capable_as_USB_Device`
    - 3. The check fails if B29...27 does not match VIF field `Product_Type`
    - 4. The check fails if B26 does not match VIF field `Modal_Operation_Supported`
    - 5. In PD2, the check fails if B25...16 is not set to zero.
    - 6. In PD3, the check fails if B25...23 is not set to zero.
    - 7. In PD3, the check fails if B22...21 does not match VIF field `ID_Header_Connector_Type`
    - 8. In PD3, the check fails if B20...16 is not set to zero.
    - 9. The check fails if B15...0 does not match VIF field `USB_VID`
  - iv. The check fails if Cert Stat VDO does not match VIF field `XID`.
  - v. The check fails if Product VDO does not match VIF fields `PID` and `bcdDevice`
  - vi. Passive Cable VDO consistency with respective VIF fields:
    - 1. B31...28: `Cable_HW_Vers`
    - 2. B27...24: `Cable_FW_Vers`
    - 3. B23...21:
      - a. In PD2 mode: Reserved 000b
      - b. In PD3 mode: VDO Version is 000b.
    - 4. B20: Reserved 0b
    - 5. B19...18:
      - a. In PD2 mode: `Type_C_To_Type_A_B_C`
      - b. In PD3 mode: `Type_C_To_Type_C_Capt_Vdm_V2`
    - 6. B17:
      - a. In PD2 mode: (Reserved) 0b
      - b. In PD3 mode: `EPR_Mode_Capable`
    - 7. B16...13 `Cable_Latency`

8. B12...11 Cable\_Termination\_Type
9. In PD2 mode: B10...7: Any Value Permitted
10. In PD3 mode
  - a. B10...9: Max\_VBUS\_Voltage
  - b. B10...9: If EPR Mode Capable is set to 1b, check that Max\_VBUS\_Voltage is set to 11b(50V)
  - c. B8...7: Reserved 00b
11. B6...5
  - a. In PD2: VBUS\_through\_cable is YES: Cable\_VBUS\_Current
  - b. In PD2: VBUS\_through\_cable is NO: (Reserved) 00b
  - c. In PD3: Cable\_VBUS\_Current
  - d. In PD3: If EPR Mode Capable is set to 1b, check that Cable\_VBUS\_Current is set to 10b(5A)
12. B4
  - a. In PD2: VBUS\_through\_cable
  - b. In PD3: (Reserved) 0b
13. B3 (Reserved) 0b
14. B2...0 Cable\_Superspeed\_Support (in PD2 only)
15. B2...0 Cable\_USB\_Highest\_Speed (in PD3 only)
- vii. Active Cable VDO consistency (in PD2) or Active Cable VDO 1 consistency (in PD3) with respective VIF fields
  1. B31...28: Cable\_HW\_Vers
  2. B27...24: Cable\_FW\_Vers
  3. B23...21:
    - a. In PD2: Reserved 000b
    - b. In PD3: VDO Version is 011b.
  4. B20: Reserved 0b
  5. B19...18:
    - a. In PD2: Type\_C\_To\_Type\_A\_B\_C
    - b. In PD3: Type\_C\_To\_Type\_C\_Capt\_Vdm\_V2
  6. B17:
    - a. In PD2 mode: (Reserved) 0b
    - b. In PD3 mode: EPR Mode Capable
  7. B16...13 Cable\_Latency
  8. B12...11 Cable\_Termination\_Type
  9. In PD2: B10...7: Any Value Permitted
  10. In PD3:
    - a. B10...9: Max\_VBUS\_Voltage
    - b. B10..9: If EPR Mode Capable is set to 1b, check that Max\_VBUS\_Voltage is set to 11b(50V)
    - c. B8: SBU\_Supported
    - d. B7: SBU\_Active
  11. B6...5
    - a. If VBUS\_Through\_Cable is NO, Reserved 00b.
    - b. If VBUS\_Through\_Cable is YES, Cable\_VBUS\_Current

- c. In PD3: If EPR Mode Capable is set to 1b, check that Cable\_VBUS\_Current is set to 10b(5A)
  - 12. B4 VBUS\_through\_cable
  - 13. B3 Cable\_SOP\_controller
  - 14. B2...0 Cable\_Superspeed\_Support (in PD2 only)
  - 15. B2...0 Cable\_USB\_Highest\_Speed (in PD3 only)
- viii. Active Cable VDO 2 consistency in PD3
  - 1. B31...24 Act\_Cab\_Max\_Op\_Temp
  - 2. B23...16 Act\_Cab\_ShutDn\_Temp
  - 3. B15 Reserved 0b.
  - 4. B14...12 Act\_Cab\_U3\_CLd\_Power
  - 5. B11 Act\_Cab\_U3\_U0\_Trans\_Mode
  - 6. B10 Act\_Cab\_Phys\_Connect
  - 7. B9 Act\_Cab\_Active\_Elem
  - 8. B8 Act\_Cab\_USB4\_Supp
  - 9. B7...6 Act\_Cab\_USB\_2\_0\_Hub\_Hops
  - 10. B5 Act\_Cab\_USB\_2\_0\_Supp
  - 11. B4 Act\_Cab\_USB\_3\_2\_Supp
  - 12. B3 Act\_Cab\_USB\_Lanes
  - 13. B2 Act\_Cab\_Opt\_Isol
  - 14. B1 USB4 Asymmetric Mode Supported.
  - 15. B0 Act\_Cab\_USB\_Gen
- ix. VPD consistency in PD3
  - 1. B31...28 VPD\_HW\_Vers
  - 2. B27...24 VPD\_FW\_Vers
  - 3. B23...21 VDO Version is 000b.
  - 4. B20...17 Reserved 0000b.
  - 5. B16...15 VPD\_Max\_VBUS\_Voltage
  - 6. B14
    - a. VPD\_Charge\_Through\_Support = 1: VPD\_Charge\_Through\_Current
    - b. VPD\_Charge\_Through\_Support = 0: Reserved 0b
  - 7. B13 Reserved 0b.
  - 8. B12...7
    - a. VPD\_Charge\_Through\_Support = 1: VPD\_VBUS\_Impedance
    - b. VPD\_Charge\_Through\_Support = 0: Reserved 000000b
  - 9. B6...1
    - a. VPD\_Charge\_Through\_Support = 1: VPD\_Ground\_Impedance
    - b. VPD\_Charge\_Through\_Support = 0: Reserved 000000b
  - 10. B0 VPD\_Charge\_Through\_Support
- 4. The Tester sends a *Discover SVIDs* Command to the UUT, using SOP'. The Tester checks the response message: **[TEST.PD.VDM.CBL.1#2]**

- a. The check fails if the UUT responds anything but a *Discover SVIDs* Response message.
  - b. If the UUT responds with "Responder BUSY" message, the Tester resends *Discover SVIDs* Command for *nBusyCount* times every *tVDMBusy* min. Once the Tester has repeated *Discover SVIDs* for *nBusyCount* times, the check fails, and the test ends here.
  - c. If the UUT responds with a "Responder NAK" message, the check fails if VIF field Modal\_Operation\_Supported = YES. The test ends here.
  - d. If the UUT responds with a "Responder ACK" message, the check fails if VIF field Modal\_Operation\_Supported = NO.
5. The Tester evaluates the number of SVIDs in the *Discover SVIDs* ACK message and checks the following: [TEST.PD.VDM.CBL.1#3]
- a. Each discovered SVID appears in VIF CableSVIDList. SVID's position in the VDO is checked according to SVID\_Fixed value in VIF.
    - i. If SVID\_Fixed is set to YES, the order of SVIDs in the VDO matches the order of CableSVID elements in VIF CableSVIDList.
    - ii. If SVID\_Fixed is set to NO, each SVID in the VDO list can be in any position in the VIF's CableSVIDList.
  - b. The number of SVIDs discovered lies between Num\_SVIDs\_min and Num\_SVIDs\_max.
  - c. For each VDO, the Tester checks if B15...0 (SVID n+1) are set to zeros, this marks the last VDO.
  - d. For each VDO, the Tester checks if B31...16 (SVID n) are set to zeros, then B15...0 shall also be zeros, and this marks the last VDO.
  - e. If the *Discover SVIDs* ACK message has 6 VDOs, and B15...0 in VDO 6 is non-zero, then repeat Step-4 until the Tester finds the last VDO.
6. For each SVID:
- a. The Tester sends a *Discover Modes* Command to the UUT, using SOP'.
  - b. The Tester checks the response: [TEST.PD.VDM.CBL.1#4]
    - i. The check fails if the UUT responds anything but a *Discover Modes* Response message.
    - ii. If the UUT responds with "Responder BUSY" message, the Tester resends *Discover Modes* Command for *nBusyCount* times every *tVDMBusy* min. Once the Tester has repeated *Discover Modes* for *nBusyCount* times, the check fails, and the test ends here.
    - iii. If the UUT responds with a "Responder ACK" message, the check fails if VIF field Modal\_Operation\_Supported = NO.
    - iv. The number of Modes discovered lies between SVID\_Num\_Modes\_Min and SVID\_Num\_Modes\_Max.
    - v. If the UUT responds with a "Responder NAK" message, the check fails if VIF field Modal\_Operation\_Supported = YES. The test ends here.
    - vi. Check that each Mode discovered matches a CableSVIDMode element in the VIF CableSVIDModeList The procedure for matching them is as follows:
      1. If SVID\_Modes\_Fixed is set to YES, the match is done by simply comparing the value in the VDO with the corresponding value in

- the SVID\_Mode\_Recog\_Value. Both lists should be in the same sequence.
2. If SVID\_Modes\_Fixed is set to NO, the value in the VDO is bitwise ANDed with each SVID\_Mode\_Recog\_Mask and the result is compared with the corresponding SVID\_Mode\_Recog\_Value in turn until a match is found.
7. For every Mode in each SVID:
    - a. The Tester sends *Enter Mode* to the UUT, using SOP'. If the corresponding VIF CableSVIDMode element contains SVID\_Mode\_VDO, the tester shall append SVID\_Mode\_VDO to the VDM.
    - b. The Tester checks the response: [TEST.PD.VDM.CBL.1#5]
      - i. If the UUT responds with "Responder BUSY", the check fails.
      - ii. If the UUT responds with "Responder NAK" the text continues.
      - iii. If the UUT responds with "Responder ACK" to *Enter Mode*, the Tester then sends an *Exit Mode* Command to the UUT using SOP'. The check fails if the UUT does not respond with "Responder ACK" to *Exit Mode*.
  8. The Tester sends *Attention* to the UUT using SOP'. The check fails if the UUT does not ignore the message. [TEST.PD.VDM.CBL.1#6]

#### 5.7.2 PD3 Mode

The tests in this subsection are run only in PD3 mode.

The tests in this subsection are applicable to UUT with VIF field VIF\_Product\_Type set to Cable.

##### TEST.PD.VDM.CBL3.1 Revision Number Test

**Description:** As a Source, the Tester sends *Discover Identity* Command with the Message Header containing a higher revision number than that supported and the UUT responds using the highest supported revision number.

##### Test Conditions:

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	VPD, Captive Cable, Si-Only eMarker
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

##### Test Procedures:

1. There are 4 possible bring-up procedures:
  - a. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3

- b. The UUT has VIF field VIF\_Product\_Type set to Port Product and PD\_Port\_Type set to eMarker. The Tester runs bring-up procedure with the UUT as a VPD COMMON.PROC.BU.10
  - c. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - d. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sets the Specification Revision (in the Message Header) to 11b and transmits a *Discover Identity* Command to the UUT, using SOP'.
  3. The Tester checks the response: **[TEST.PD.VDM.CBL3.1#1]**
    - a. The check fails if either “Responder ACK” or “Responder BUSY” message is not sent before *tVDMReceiverResponse*. This delay is from the last bit of the EOP of the *GoodCRC* Message of the *Discover Identity* Command to the first bit of the response.
    - b. The Tester checks that the Specification Revision (in the Message Header) is set consistent with VIF field PD\_Specification\_Revision.
    - c. If the UUT responds with a BUSY message, the Tester resends *Discover Identity* Command for *nBusyCount* times every *tVDMBusy* min. Once the Tester has repeated *Discover Identity* for *nBusyCount* times, the check fails if the UUT does not respond with a “Responder ACK” message.

## 6 Power Supply Specific Tests

Table 14 PDO Sequence

Number of PDOs	1	2	3	4	5	6	7
Transition #1	1 to 1	1 to 2	1 to 2	1 to 2	1 to 2	1 to 2	1 to 2
Transition #2		2 to 1	2 to 1	2 to 1	2 to 1	2 to 1	2 to 1
Transition #3			1 to 3	1 to 3	1 to 3	1 to 3	1 to 3
Transition #4			3 to 2	3 to 2	3 to 2	3 to 2	3 to 2
Transition #5			2 to 3	2 to 3	2 to 3	2 to 3	2 to 3
Transition #6			3 to 1	3 to 1	3 to 1	3 to 1	3 to 1
Transition #7				1 to 4	1 to 4	1 to 4	1 to 4
Transition #8				4 to 3	4 to 3	4 to 3	4 to 3
Transition #9				3 to 4	3 to 4	3 to 4	3 to 4
Transition #10				4 to 2	4 to 2	4 to 2	4 to 2
Transition #11				2 to 4	2 to 4	2 to 4	2 to 4
Transition #12				4 to 1	4 to 1	4 to 1	4 to 1
Transition #13					1 to 5	1 to 5	1 to 5
Transition #14					5 to 4	5 to 4	5 to 4
Transition #15					4 to 5	4 to 5	4 to 5
Transition #16					5 to 3	5 to 3	5 to 3
Transition #17					3 to 5	3 to 5	3 to 5
Transition #18					5 to 2	5 to 2	5 to 2
Transition #19					2 to 5	2 to 5	2 to 5
Transition #20					5 to 1	5 to 1	5 to 1
Transition #21						1 to 6	1 to 6
Transition #22						6 to 5	6 to 5
Transition #23						5 to 6	5 to 6
Transition #24						6 to 4	6 to 4
Transition #25						4 to 6	4 to 6
Transition #26						6 to 3	6 to 3
Transition #27						3 to 6	3 to 6
Transition #28						6 to 2	6 to 2
Transition #29						2 to 6	2 to 6
Transition #30						6 to 1	6 to 1
Transition #31							1 to 7
Transition #32							7 to 6
Transition #33							6 to 7
Transition #34							7 to 5
Transition #35							5 to 7
Transition #36							7 to 4
Transition #37							4 to 7
Transition #38							7 to 3
Transition #39							3 to 7
Transition #40							7 to 2
Transition #41							2 to 7
Transition #42							7 to 1

Table 15 EPR PDO Sequence



Number of PDOs	8	9	10	11
Transition #1	1 to 8	1 to 8	1 to 8	1 to 8
Transition #2	8 to 7	8 to 7	8 to 7	8 to 7
Transition #3	7 to 8	7 to 8	7 to 8	7 to 8
Transition #4	8 to 6	8 to 6	8 to 6	8 to 6
Transition #5	6 to 8	6 to 8	6 to 8	6 to 8
Transition #6	8 to 5	8 to 5	8 to 5	8 to 5
Transition #7	5 to 8	5 to 8	5 to 8	5 to 8
Transition #8	8 to 4	8 to 4	8 to 4	8 to 4
Transition #9	4 to 8	4 to 8	4 to 8	4 to 8
Transition #10	8 to 3	8 to 3	8 to 3	8 to 3
Transition #11	3 to 8	3 to 8	3 to 8	3 to 8
Transition #12	8 to 2	8 to 2	8 to 2	8 to 2
Transition #13	2 to 8	2 to 8	2 to 8	2 to 8
Transition #14	8 to 1	8 to 1	8 to 1	8 to 1
Transition #15		1 to 9	1 to 9	1 to 9
Transition #16		9 to 8	9 to 8	9 to 8
Transition #17		8 to 9	8 to 9	8 to 9
Transition #18		9 to 7	9 to 7	9 to 7
Transition #19		7 to 9	7 to 9	7 to 9
Transition #20		9 to 6	9 to 6	9 to 6
Transition #21		6 to 9	6 to 9	6 to 9
Transition #22		9 to 5	9 to 5	9 to 5
Transition #23		5 to 9	5 to 9	5 to 9
Transition #24		9 to 4	9 to 4	9 to 4
Transition #25		4 to 9	4 to 9	4 to 9
Transition #26		9 to 3	9 to 3	9 to 3
Transition #27		3 to 9	3 to 9	3 to 9
Transition #28		9 to 2	9 to 2	9 to 2
Transition #29		2 to 9	2 to 9	2 to 9
Transition #30		9 to 1	9 to 1	9 to 1
Transition #31			1 to 10	1 to 10
Transition #32			10 to 9	10 to 9
Transition #33			9 to 10	9 to 10
Transition #34			10 to 8	10 to 8
Transition #35			8 to 10	8 to 10
Transition #36			10 to 7	10 to 7
Transition #37			7 to 10	7 to 10
Transition #38			10 to 6	10 to 6
Transition #39			6 to 10	6 to 10
Transition #40			10 to 5	10 to 5
Transition #41			5 to 10	5 to 10
Transition #42			10 to 4	10 to 4
Transition #43			4 to 10	4 to 10
Transition #44			10 to 3	10 to 3
Transition #45			3 to 10	3 to 10
Transition #46			10 to 2	10 to 2
Transition #47			2 to 10	2 to 10
Transition #48			10 to 1	10 to 1
Transition #49				1 to 11

Transition #50				11 to 10
Transition #51				10 to 11
Transition #52				11 to 9
Transition #53				9 to 11
Transition #54				11 to 8
Transition #55				8 to 11
Transition #56				11 to 7
Transition #57				7 to 11
Transition #58				11 to 6
Transition #59				6 to 11
Transition #60				11 to 5
Transition #61				5 to 11
Transition #62				11 to 4
Transition #63				4 to 11
Transition #64				11 to 3
Transition #65				3 to 11
Transition #66				11 to 2
Transition #67				2 to 11
Transition #68				11 to 1

## 6.1 Test Procedures for Source Capable UUT

### 6.1.1 PD2 and PD3 Modes

The tests in this subsection are run firstly in PD2 mode and secondly in PD3 mode.

The tests in this subsection are applicable to UUT with VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer and DRP.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (ie, a Provider/Consumer or Consumer/Provider) The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### TEST.PD.PS.SRC.1 Multiple Request Load Test

Description: As a Sink only, the Tester verifies that the Source UUT responds correctly to various *Request* Messages.

Test Specific Tester Behavior: N/A

Test Conditions:

	Consumer Only, C/P	Provider Only, P/C, DRP
<i>Rev2Src</i>		✓
<i>Rev3ChkdSrc</i>		✓

Test Requirements: If Captive\_Cable = YES, *vSrcNew* min, *vAvsNewmin*, or *vPpsNew* min limit shall be (*vSrcNew* or *vAvsNew* min) - (0.75\*I/X) or (*vPpsNew* min) - (0.75\*I/X), where I is the actual current being drawn, and X=3 if the sourced PDO contract is within the range (0, 3A], or X=5 if the sourced PDO contract is within the range (3A, 5A)

Test Procedures:

1. The Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester repeats the following steps for each of the PDOs (specified in the VIF), starting with the first PDO:
  - a. If the next Request will be for a PPS transition involving a current decrease (but no PDO change), the Tester reduces its current draw to the new level required, before sending the Request. (This avoids the current limit in the UUT shutting off  $V_{BUS}$ .) The Tester draws the requested Operating Current / Power with a current transition slew rate of 100mA per  $\mu$ s. The Tester monitors  $V_{BUS}$  voltage and the check fails if  $V_{BUS}$  voltage is not within the limits of *vPpsNew*.

[TEST.PD.PS.SRC.1#9]

NOTE: All measurements shall be within tolerances defined in the main specification and depicted in **Figure 6** Voltage and Current Tolerances

- b. The Tester defines *Request* Message as following:
  - i. B27 (Reserved) = 0b
  - ii. B26 (Capability Mismatch) = 0b
  - iii. B25 (USB Communication Capable) = 0b
  - iv. B24 (No USB Suspend) = 1b
  - v. B22 (EPR Capable) = 0b
  - vi. For Fixed, Battery or Variable PDO:
    1. B19...10 (Operating Current / Power):
      - a. For first Request: 0mA/0mW
      - b. For subsequent Request: As defined in Step 2.f.i
    2. B9...0 (Maximum Operating Current / Power) = Maximum Current / Power in the Source PDO.
  - vii. For PPS APDO:
    1. B20...9 (Output Voltage) = Maximum Voltage offered in Source APDO
    2. B6...0 (Operating Current):
      - a. For first Request: 1A
      - b. For subsequent Request: As defined in Step 2.f.ii
  - viii. For SPR AVS APDO:
    1. B20...9 (Output Voltage) – there are three iterations:
      - a. 9V (minimum)
      - b. 13.8V (an arbitrary value near the middle of the range)
      - c. 20V if available, otherwise 15V (maximum)
    2. B6...0 (Operating Current):
      - a. For first Request: 0mA/0mW

- b. For subsequent Request: As defined in Step 2.f.i
- The initial Request is for PDO #1 (5V Fixed) at 0mA. The complete list of Requests is defined in step 2.f.
- c. If VIF parameter Has\_Invariant\_PDOs is set to Y, or the current Source Capabilities match those in the VIF:
  - i. The Tester sends *Request* Message defined above.
  - ii. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
  - iii. If an *Accept* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SRC.1#10]
  - iv. If a *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [TEST.PD.PS.SRC.1#11]
- d. Otherwise, If VIF parameter Has\_Invariant\_PDOs is set to N, and the current Source Capabilities do not match those in the VIF:
  - i. The Tester repeats sending the previous *Request* Message (for what is currently contracted) but with the Mismatch bit set (this prevents interfering with power test procedure, by not changing the current contract)
  - ii. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
  - iii. If an *Accept* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SRC.1#10]
  - iv. If the first Preamble bit of *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [TEST.PD.PS.SRC.1#11]
  - v. The Tester responds to any *Get\_Sink\_Cap* Message with *Sink\_Capabilities* Message using the values found in the UUT VIF Source PDO values. It also responds to any *Get\_Sink\_Cap\_Extended* Message using the values defined in the table below (see Table 16 Sink Capabilities Extended Message).
  - vi. If Source Capabilities are received with less than the Sink Capabilities (same as VIF Source Capabilities) and *tCapabilitiesMismatchResponse* have not elapsed, then the Tester goes back and repeats the *Request* Message in step I with the Mismatch bit set.
  - vii. Within *tCapabilitiesMismatchResponse*
    - 1. If the *Source Capabilities* matching those in the VIF have been received by the Tester, it sends the *Request* Message defined above with B19..10(Operating Current) replaced with Maximum Current in Source PDO. The Tester passes the check and proceeds to the next step. [TEST.PD.PS.SRC.1#12]

2. Else, run **Check 1** and **Check 2** from COMMON.PROC.PD.18  
[TEST.PD.PS.SRC.1#14]
- viii. If *Wait* Message is received, the Tester re-sends the *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
- ix. If *Accept* Message is not received, the check fails, and the test stops here.
- x. If the first Preamble bit of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [TEST.PD.PS.SRC.1#13]
- e. The Tester performs the following checks:
  - i. If the transition involves a current decrease (but no PDO change)  
[TEST.PD.PS.SRC.1#1]
    1. For Fixed, Battery or Variable PDOs:  
The Tester decreases current to the new value within *tSnkNewPower* min of the last bit of the *GoodCRC* sent in response to the *Accept* Message. The Tester draws the requested Operating Current / Power with a current transition slew rate of 100mA per  $\mu$ s. The Tester monitors  $V_{BUS}$  voltage and the check fails if  $V_{BUS}$  voltage is not within the limits of *vSrcNew*.
    2. For PPS:  
The current was already reduced in step 2.a.
  - ii. If the transition involves a current increase (but no PDO change), the Tester increases current to the new value after the last bit of the *GoodCRC* sent in response to the *PS\_RDY* Message. The Tester draws the requested Operating Current / Power with a current transition slew rate of 100mA per  $\mu$ s. The Tester monitors  $V_{BUS}$  voltage and the check fails if  $V_{BUS}$  voltage is not within the limits of *vSrcNew*, *vAvsNew* or *vPpsNew*.  
[TEST.PD.PS.SRC.1#2]
  - iii. If the transition involves a change of PDO with a voltage increase, decrease the current drawn by the Tester to less than *iSnkStdby* within *tSnkStdby* of the last bit of the *GoodCRC* sent in response to the *Accept* Message. If the transition involves a change of PDO with decreasing voltage and current, the Tester decreases its current draw to the requested current or power for Fixed, Battery, Variable and SPR\_AVs PDOs, 80% of the requested current for PPS PDO within *tSrcTransition* min after the end of the *GoodCRC* sent in response to the *Accept* Message. The Tester performs the following checks:
    1. The Tester checks that  $V_{BUS}$  remains within *vSrcNew*, *vAvsNew* or *vPpsNew* of the starting voltage within *tSrcTransition* min after the end of the *GoodCRC* sent in response to the *Accept* Message.  
[TEST.PD.PS.SRC.1#3]

2. The Tester checks that the  $V_{BUS}$  transition meets *vSrcSlewPos* and *vSrcSlewNeg*. (See [Appendix E. Slew Rate Measurements.](#) )  
[TEST.PD.PS.SRC.1#4]
3. During the voltage transition, the Tester checks that the  $V_{BUS}$  voltage remains within *vSrcValid* limits from the time of crossing into the *vSrcValid* limits until *tSrcSettle* max.  
[TEST.PD.PS.SRC.1#5]
4. The Tester checks that the  $V_{BUS}$  voltage remains within *vSrcNew*, *vAvsNew* or *vPpsNew* limits between *tSrcSettle* max and 100ms after *tSrcSettle* max. [TEST.PD.PS.SRC.1#6]
5. The Tester checks that the UUT does not send *PS\_RDY* before  $V_{BUS}$  is within *vSrcNew*, *vAvsNew* or *vPpsNew*.  
[TEST.PD.PS.SRC.1#7]
6. After having sent *GoodCRC* in response to *PS\_RDY*, if the transition involves a current increase, the Tester increases its current draw to:
  - a. For Fixed, Battery, Variable PDOs and SPR\_AVs PDOs:  
The current or power requested.
  - b. For PPS PDO:  
80% of the current requested (except that for the last step 0mA will be drawn.  
The Tester checks that the  $V_{BUS}$  voltage remains within *vSrcNew*, *vAvsNew* or *vPpsNew* limits for 100ms after receiving *PS\_RDY*[TEST.PD.PS.SRC.1#8]
- f. The Tester repeats Step-2a to Step-2e, replacing the appropriate fields with the next value from the following list. The Tester sends 9 separate *Request* Messages for each PDO, 7 separate *Request* Messages for PPS, and 9 separate *Request* Messages for each SPR\_AVs voltage iteration:
  - i. For Fixed, Battery and Variable PDOs, replace B19...10 (Operating Current / Power). For SPR AVs, replace B6...0 (Operating Current). All 9 Request Messages are sent for each voltage iteration before advancing to the next one.
    1. 0mA / 0mW [performed in Step 2.b; listed here for reference only]
    2. 25% of Maximum Current / Power in the Source PDO
    3. 50% of Maximum Current / Power in the Source PDO
    4. 75% of Maximum Current / Power in the Source PDO
    5. 100% of Maximum Current / Power in the Source PDO
    6. 75% of Maximum Current / Power in the Source PDO
    7. 50% of Maximum Current / Power in the Source PDO
    8. 25% of Maximum Current / Power in the Source PDO
    9. 0mA / 0mW
  - ii. For PPS APDO, replace B20...9 (Output Voltage) and B6...0 (Operating Current):

1. Maximum Voltage offered in Source APDO @ 1A (Tester will draw 0mA) [performed in Step 2.b; listed here for reference only]
2. Maximum Voltage offered in Source APDO @ 1A (Tester will draw 80% of this.)
3. Maximum Voltage offered in Source APDO @  $(1A + \text{Maximum Current offered in Source APDO})/2$ . (Tester will draw 80% of this.)
4. Maximum Voltage offered in Source APDO @ Maximum Current offered in Source APDO. (Tester will draw 80% of this.)
5. Maximum Voltage offered in Source APDO @  $(1A + \text{Maximum Current offered in Source APDO})/2$ . (Tester will draw 80% of this.)
6. Maximum Voltage offered in Source APDO @ 1A (Tester will draw 80% of this)
7. Maximum Voltage offered in Source APDO @ 1A (Tester will draw 0mA)

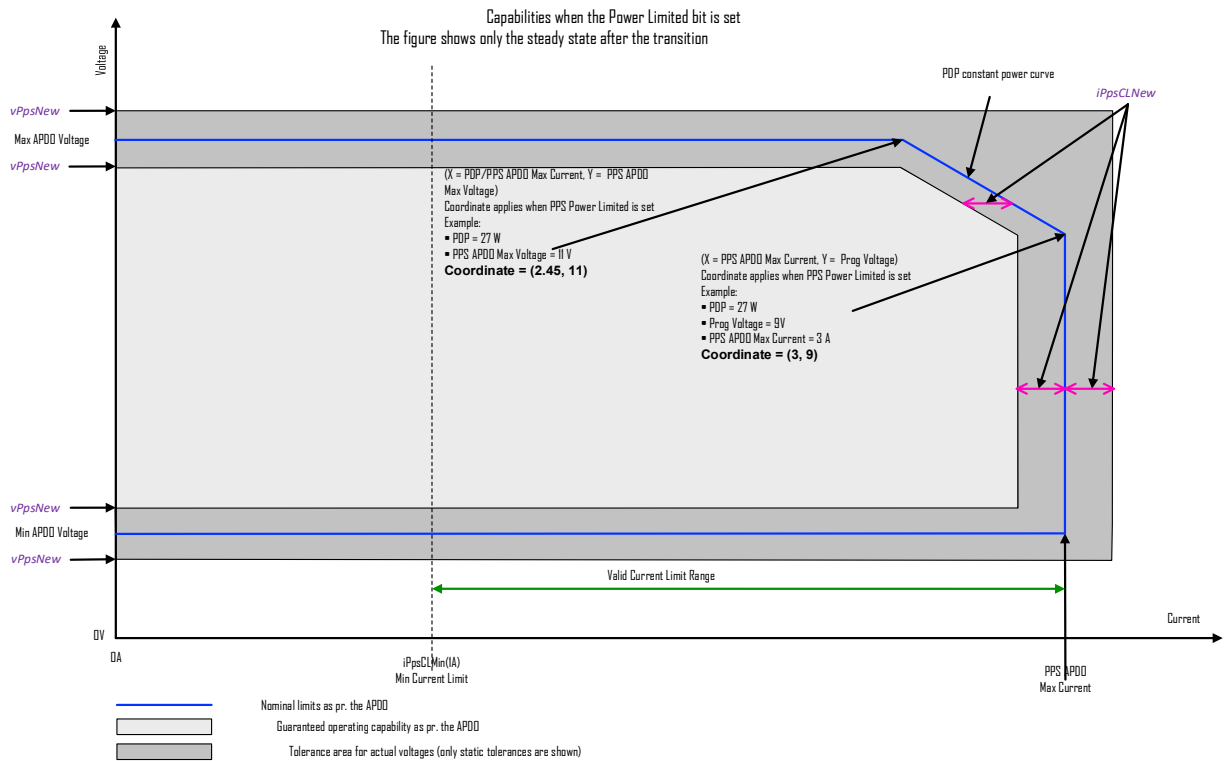


Figure 6 Voltage and Current Tolerances

Table 16 Sink Capabilities Extended Message

Field	Description
-------	-------------



<b>VID</b>	0x1A0A
<b>PID</b>	0x0302
<b>XID</b>	0
<b>FW Version</b>	0
<b>HW Version</b>	0
<b>SKEDB Version</b>	1
<b>Load Step</b>	0
<b>Sink Load Characteristics</b>	0
<b>Compliance</b>	0
<b>Touch Temp</b>	0
<b>Battery Info</b>	0
<b>Sink Modes</b>	Bit 0: 1 Bit 1: 1 Bit 2: 0 Bit 3: 0 Bit 4: 0 Bit 5: 1 Bit 6: 0 Bit 7: 0
<b>Sink Minimum PDP</b>	Copy from VIF PD_Power_As_Source_Guaranteed, maximum of 100
<b>Sink Operational PDP</b>	Copy from VIF PD_Power_As_Source (Maximum PDP), maximum of 100
<b>Sink Maximum PDP</b>	Copy from VIF PD_Power_As_Source (Maximum PDP), maximum of 100
<b>EPR Sink Minimum PDP</b>	0
<b>EPR Sink Operational PDP</b>	0
<b>EPR Sink Maximum PDP</b>	0

#### *TEST.PD.PS.SRC.2 PDO Transitions*

**Description:** As a Sink only, the Tester enumerates all possible transitions among the PDOs of the UUT and verifies the UUT performs PDO transitions correctly.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, C/P	Provider Only, P/C, DRP
<i>Rev2Src</i>		✓
<i>Rev3ChkdSrc</i>		✓

**Test Procedures:**

1. The Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.

2. The Tester repeats the following steps using the PDO as provided in the PDO Sequence Table (see [Table 14](#)), based on the Source PDOs specified in the VIF, and for PPS and AVS APDO, request the max, min, and mid voltages in sequence:
  - a. The Tester defines *Request* Message as following:
    - i. B27 (Reserved) = 0b
    - ii. B26 (Capability Mismatch) = 0b
    - iii. B25 (USB Communication Capable) = 0b
    - iv. B24 (No USB Suspend) = 1b
    - v. For Fixed, Battery or Variable PDO:
 

B19...10 (Operating Current / Power) = 100mA / 0.5W

B9...0 (Maximum Operating Current / Power) = 100mA / 0.5W

For PPS APDO:

      - a. B20...9 (Output Voltage) – there are at least three iterations:
        - i. 5V (minimum)
        - ii. 6V (small step)
        - iii. 10V (maximum and last iteration for 9V Prog)
        - iv. 20V if available, otherwise 15V (maximum)

B6...0 (Operating Current) = 1A

For AVS APDO:

B20...9 (Output Voltage), Iterate though Max, Min and Mid Voltages offered in Source APDO.

      - b. 20V if available otherwise 15V (maximum)
      - c. 9V (minimum)
      - d. 10V (small step)
      - e. 12.5V (an arbitrary value near the middle of the range for this test)

B6...0 (Operating Current) = 1A
  - b. If VIF parameter Has\_Invariant\_PDOs is set to Y, or the current Source Capabilities match those in the VIF:
    - i. The Tester sends *Request* Message defined above.
    - ii. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
    - iii. If an *Accept* Message is not received, the check fails, and the test stops here. [\[TEST.PD.PS.SRC.2#1\]](#)
    - iv. If the first Preamble bit of *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [\[TEST.PD.PS.SRC.2#2\]](#)
  - c. Otherwise, If VIF parameter Has\_Invariant\_PDOs is set to N, and the current Source Capabilities do not match those in the VIF:
    - i. The Tester repeats sending the previous *Request* Message (for what is currently contracted) but with the Mismatch bit set (this prevents

- interfering with power test procedure, by not changing the current contract)
- ii. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
  - iii. If an *Accept* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SRC.2#1]
  - iv. If a *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [TEST.PD.PS.SRC.2#2]
  - v. The Tester responds to any *Get\_Sink\_Cap* Message with *Sink\_Capabilities* Message using the values found in the UUT VIF Source PDO values. It also responds to any *Get\_Sink\_Cap\_Extended* Message using the values defined in the table (see Table 16).
  - vi. If Source Capabilities are received with less than the Sink Capabilities (same as VIF Source Capabilities) and *tCapabilitiesMismatchResponse* has not elapsed, then the Tester goes back and repeats the *Request* Message in step I with the Mismatch bit set.
  - vii. Within *tCapabilitiesMismatchResponse*,
    1. If the Source Capabilities matching those in the VIF have been received by the Tester, the Tester sends the *Request* Message defined above with B19..10 (Operating Current) replaced with Maximum Current in Source PDO. The Tester passes the check and proceeds to the next step. [TEST.PD.PS.SRC.2#18]
    2. Else, run Check 1 and Check 2 from COMMON.PROC.PD.18. [TEST.PD.PS.SRC.2#20]
  - viii. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
  - ix. If *Accept* Message is not received, the check fails, and the test stops here.
  - x. If the first Preamble bit of *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [TEST.PD.PS.SRC.2#19]
- d. The Tester performs the following checks:
- i. For transitions to a new PDO: Decrease the current drawn by the Tester to less than *iSnkStdby* within *tSnkStdby* min of the last bit of the *GoodCRC* sent in response to the *Accept* Message. The Tester performs the following checks:
    1. The Tester checks the  $V_{BUS}$  is within *vSrcNew* or *vPpsNew* or *vAvsNew* of the starting voltage within *tSrcTransition* min after the end of the *GoodCRC* sent in response to the *Accept* Message. [TEST.PD.PS.SRC.2#3]

2. The Tester checks that the  $V_{BUS}$  transition meets  $vSrcSlewPos$  and  $vSrcSlewNeg$ . [TEST.PD.PS.SRC.2#4] (see [Appendix E. Slew Rate Measurements.](#) )
  3. During the voltage transition, the Tester checks that the  $V_{BUS}$  voltage remains within  $vSrcValid$  limits from the time of crossing into the  $vSrcValid$  limits until  $tSrcSettle$  max. [TEST.PD.PS.SRC.2#5]
  4. The Tester checks that the  $V_{BUS}$  voltage is within  $vSrcNew$ ,  $vPpsNew$  or  $vAvsNew$  limits between  $tSrcSettle$  max and 100ms after  $tSrcSettle$  max. [TEST.PD.PS.SRC.2#6]
  5. The Tester checks that the UUT does not send  $PS\_RDY$  before the  $V_{BUS}$  is within  $vSrcNew$ ,  $vPpsNew$  or  $vAvsNew$ . [TEST.PD.PS.SRC.2#7]
- ii. For transitions within PPS PDOs:
1. The Tester checks that the  $V_{BUS}$  transition meets  $vPpsSlewPos$  and  $vPpsSlewNeg$ , (see [Appendix E. Slew Rate Measurements.](#)) [TEST.PD.PS.SRC.2#8]
  2. If the voltage step  $\leq vPpsSmallStep$ , the Tester checks that  $V_{BUS}$  settles within the limits defined by  $vPpsNew$  within  $tPpsTransSmall$  after the last bit of the *GoodCRC* sent in response to the *Accept* Message. [TEST.PD.PS.SRC.2#9]
  3. If the voltage step  $> vPpsSmallStep$ , and PPS is not in the CL mode, the Tester checks that  $V_{BUS}$  settles within the limits defined by  $vPpsNew$  and that  $PS\_RDY$  is sent within  $tPpsTransLarge$  after the last bit of the *GoodCRC* sent in response to the *Accept* Message. [TEST.PD.PS.SRC.2#10]
  4. During the voltage transition the Tester checks that any  $V_{BUS}$  overshoot or undershoot beyond  $vPpsNew$  does not exceed  $vPpsValid$  limits. [TEST.PD.PS.SRC.2#11]
  5. The Tester checks that the UUT does not send  $PS\_RDY$  before the  $V_{BUS}$  is within  $vPpsNew$ . [TEST.PD.PS.SRC.2#12]
- iii. For transitions within AVS APDOs:
1. The Tester checks that the  $V_{BUS}$  transition meets  $vAvsSlewPos$  and  $vAvsSlewNeg$ , (see [Appendix E. Slew Rate Measurements.](#)) [TEST.PD.PS.SRC.2#13]
  2. If the voltage step  $\leq vAvsSmallStep$ , the Tester checks that  $V_{BUS}$  settles within the limits defined by  $vAvsNew$  and that  $PS\_RDY$  is sent within  $tAvsTransSmall$  after the last bit of the *GoodCRC* sent in response to the *Accept* Message, provided the voltage on  $V_{BUS}$  has reached  $vAvsNew$ . [TEST.PD.PS.SRC.2#14]
  3. If the voltage step  $> vAvsSmallStep$ , the Tester checks that  $V_{BUS}$  settles within the limits defined by  $vAvsNew$  and that  $PS\_RDY$  is sent  $tAvsTransLarge$  after the last bit of the *GoodCRC* sent in response to the *Accept* Message. [TEST.PD.PS.SRC.2#15]

4. During the voltage transition the Tester checks that any  $V_{BUS}$  overshoot or undershoot beyond  $vAvsNew$  does not exceed  $vAvsValid$  limits. [TEST.PD.PS.SRC.2#16]
5. The Tester checks that the UUT does not send  $PS\_RDY$  before the  $V_{BUS}$  is within  $vAvsNew$ . [TEST.PD.PS.SRC.2#17]

### TEST.PD.PS.SRC.3 Initial Source PDO Transition Post PR Swap

**Description:** The Tester verifies that after a *PR\_Swap*, the new Sink UUT responds correctly to a new *Source\_Capabilities* Messages

#### Test Specific Tester Behavior:

- The Tester is a Dual-Role Power, initially a Sink.
- The Tester sends *Sink\_Capabilities* setting B27 (Unconstrained Power) to 1b COMMON.PROC.PD.3

#### Test Conditions:

	Consumer Only, C/P	Provider Only, P/C, DRP
<i>Rev2Src</i>		✓
<i>Rev3ChkdSrc</i>		✓

#### Test Procedures:

1. The Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. After this point the tester should reject any *PR\_Swap* request.
3. The Tester sends a *PR\_Swap* Message.
4. The Tester checks the response. [TEST.PD.PS.SRC.3#1]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *PR\_Swap* every *tPRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. Once the Tester has resent *PR\_Swap* for *nCtsWaitCount* times, the check fails, and the test ends here.
  - b. If the UUT responds with *Reject* or *Not\_Supported* check VIF field PD\_Port\_Type is set to “Provider Only” or VIF field Accepts\_PR\_Swap\_As\_Src = No. The test ends here.
  - c. If the UUT responds with an *Accept* Message, check that the VIF field PD\_Port\_Type is not set to “Provider Only” and VIF field Accepts\_PR\_Swap\_As\_Src = Yes.
5. After this point the Tester shall *Reject* any *PR\_Swap* request.
6. The Tester draws no more than *iSwapStandby* within *tSrcTransition* min after it has sent the last *GoodCRC*.
7. The Tester checks that the UUT sends a *PS\_RDY* Message after the UUT driving V<sub>BUS</sub> voltage to *vSafe0V*. [TEST.PD.PS.SRC.3#2]
8. The Tester checks that the UUT sends *PS\_RDY* within *tSrcTransition* max plus *tSrcSwapStdby* max after the last bit of *GoodCRC* was sent in response to the received *Accept*. [TEST.PD.PS.SRC.3#3]
9. The Tester presents *vSafe5V* on V<sub>BUS</sub> and then it sends a *PS\_RDY* Message, both within *tNewSRC* max of the last *GoodCRC* sent by the UUT.
10. The Tester checks that the UUT draws *iSnkStby* within *tNewSnk*. [TEST.PD.PS.SRC.3#19]
11. The Tester transmits a *Source\_Capabilities* Message with the Capabilities as defined in COMMON.PROC.PD.2 Paragraph 2c. (as these will have been the *Source\_Capabilities* returned if the UUT has requested them during the

- PR\_Swap*. The Tester checks that the UUT sends a *Request* Message.  
[TEST.PD.PS.SRC.3#4]
12. The UUT is now a Sink, and the Tester is a Source.
  13. The Tester sends an *Accept* Message.
    - a. If a new PDO has been requested and the transition involves increasing the voltage:
      - i. After a total of *tSrcTransition* min, check that current drawn by the UUT does not exceed *iSnkStdby* max. [TEST.PD.PS.SRC.3#5]
      - ii. Wait *tSrcTransition* from the *GoodCRC* Message and then change  $V_{BUS}$  voltage requested to the new value and/or be prepared to supply more or less current within *tSrcTransition* max plus *tSrcReady* max as appropriate.
      - iii. Check that the average current drawn by the UUT, measured in the 1ms before sending *PS\_RDY*, does not exceed *iSnkStdby* max.  
[TEST.PD.PS.SRC.3#6]
    - b. If a new PDO with no change in the voltage or no new PDO requested, After *tSrcTransition* max + *tSrcReady* max (315ms) time period check that the current drawn from  $V_{BUS}$  does not exceed the previously contracted current. [TEST.PD.PS.SRC.3#7]
  14. The Tester sends a *PS\_RDY* Message to the UUT.
  15. The Tester performs the following check [TEST.PD.PS.SRC.3#8]
    - a. For Fixed, Battery or Variable PDO:  
The Tester checks that the current drawn by the UUT does not exceed the level specified in the current PDO over the next 5 seconds.
    - b. For PPS PDO:  
N/A
  16. The Tester presents *SinkTxOk* if the test is in PD3 mode. The Tester waits *tCtsAfterLastPoint* to respond to messages from the UUT.
  17. The Tester sends a new *Source\_Capabilities* Message with a single PDO.  
The PDO:
    - i. B31...30 (Fixed Supply) set to 00b.
    - ii. B29 (Dual-Role Power) set to 1b.
    - iii. B28 (USB Suspend Supported) set to 0b.
    - iv. B26 (USB Communications Capable) set to 0b.
    - v. B25 (Dual-Role Data) set to 1b.
    - vi. B24...22 (Reserved) set to 00b.
    - vii. B21...20 (Peak Current) set to 00b.
    - viii. B19...10 (Voltage) set to 5V.
    - ix. B9...0 (Maximum Current) set to the full requested current as provided in the 1<sup>st</sup> PDO of the *Sink\_Capabilities* from the UUT
  18. The check fails if the UUT does not responds with a *Request* Message.  
[TEST.PD.PS.SRC.3#9]
  19. The Tester sends an *Accept* Message.
    - a. After *tSrcTransition* max + *tSrcReady* max (315ms) time period check that the current drawn from  $V_{BUS}$  does not exceed the contracted current.  
[TEST.PD.PS.SRC.3#12]



- b. The Tester sends a *PS\_RDY* Message to the UUT.
20. The Tester checks that the current drawn by the UUT does not exceed the level specified in the current PDO over the next 5 seconds. [TEST.PD.PS.SRC.3#13]
21. The Tester sends a new *Source\_Capabilities* Message with a single PDO.  
The PDO:
  - i. B31...30 (Fixed Supply) set to 00b.
  - ii. B29 (Dual-Role Power) set to 1b.
  - iii. B28 (USB Suspend Supported) set to 0b.
  - iv. B26 (USB Communications Capable) set to 0b.
  - v. B25 (Dual-Role Data) set to 1b.
  - vi. B24...22 (Reserved) set to 00b.
  - vii. B21...20 (Peak Current) set to 00b.
  - viii. B19...10 (Voltage) set to 5V.
  - ix. B9...0 (Maximum Current) set to 0mA.
22. The Tester checks that it receives a *Request* Message [TEST.PD.PS.SRC.3#14]
23. The Tester sends an *Accept* Message.
24. After a total of *tSrcTransition* min, check that current drawn by the UUT does not exceed *pSnkSusp* max. [TEST.PD.PS.SRC.3#17]
25. The Tester sends a *PS\_RDY* Message to the UUT.
26. The Tester checks that the current drawn by the UUT does not exceed the level specified in the current PDO over the next 5 seconds. [TEST.PD.PS.SRC.3#18]

#### *TEST.PD.PS.SRC.4 Source Behavior with Capability Mismatch bit*

**Description:** As a Sink only, the Tester verifies that the Source UUT responds correctly to various *Request* Messages with the Capability Mismatch bit set.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only, C/P	Provider Only, P/C, DRP
<i>Rev2Src</i>		✓
<i>Rev3ChkdSrc</i>		✓

**Test Requirements:**

**Test Procedures:**

1. The Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester repeats the following steps for each of the PDOs starting with the first PDO:
  - a. The tester defines *Request* Message as follows:
    - i. B27 (Reserved) = 0b
    - ii. If Source Capabilities PDO Maximum Current < 5A or Maximum Power < 100W, B26 (Capability Mismatch) = 1b. If Source Capabilities PDO Maximum Current = 5A or Maximum Power = 100W, B26 (Capability Mismatch) = 0.

- iii. B25 (USB Communication Capable) = 0b
- iv. B24 (No USB Suspend) = 1b
- v. For Fixed, Battery or Variable PDO:
  - B19...10 (Operating Current / Power) = Maximum Current / Power in Source PDO
  - B9...0 (Maximum Operating Current / Power) = 5A /100W.
- vi. For AVS APDO:
  - B20...9 (Output Voltage) = Maximum Voltage offered in the last Source Fixed PDO.
  - B6...0 (Operating Current) = Maximum Current offered in Source AVS APDO.
  - NOTE:** In Source AVS APDO if B9...0 (Maximum Current) is set to 0, use B19...10 (Maximum Current) value.
- vii. For PPS:
  - B20...9 (Output Voltage) = Maximum Voltage offered in Source APDO
  - B6...0 (Operating Current) = Maximum current offered in Source APDO
- b. If VIF parameter Has\_Invariant\_PDOs is set to Y, or the current Source Capabilities match those in the VIF:
  - i. The Tester sends the *Request* Message defined in step 2.a.
  - ii. if a *Wait* Message is received, the Tester re-sends the *Request* every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
  - iii. If an *Accept* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SRC.4#1]
  - iv. If a *PS\_RDY* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SRC.4#2]
  - v. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message.  
[TEST.PD.PS.SRC.4#3]
- c. If VIF parameter Has\_Invariant\_PDOs is set to N, and the current Source Capabilities do not match those in the VIF:
  - i. The Tester sends the *Request* Message defined in step 2.a.
  - ii. if a *Wait* Message is received, the Tester re-sends the *Request* every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
  - iii. If an *Accept* Message is not received, the check fails, and the test stops here [TEST.PD.PS.SRC.4#4]
  - iv. If a *PS\_RDY* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SRC.4#5]

- v. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message.  
[TEST.PD.PS.SRC.4#6]
- vi. The Tester responds to any *Get\_Sink\_Cap* Message with a *Sink\_Capabilities* Message with the following PDOs:
  - a. PDO 1:
    - i. B31...30(Fixed Supply) set to 00b.
    - ii. B29 (Dual-Role Power) set to 0b.
    - iii. B28 (Higher Capacity) set to 1b
    - iv. B27 (Unconstrained Power) set to 1b.
    - v. B26 (USB Communications Capable) set to 0b.
    - vi. B25 (Dual-Role Data) set to 0b.
    - vii. B24...23 (Fast Role Swap required USB Type-C Current) set to 00b (Fast Role Swap not supported)
    - viii. B22...20 (Reserved) set to 000b.
    - ix. B19...B10(Voltage) set to 5V.
    - x. B9...0 (Maximum Current) set to 5A.
  - b. PDO 2:
    - i. B31...30(Fixed Supply) set to 00b
    - ii. B29...20 (Reserved) set to 000b
    - iii. B19...10 (Voltage) set to 20V.
    - iv. B9...0(Maximum Current) set to 5A.
  - c. PDO3:Variable 20V 5A
    - i. B31...30 (Variable Supply) set to 01b
    - ii. B29...20 (Max Voltage) set to 20V
    - iii. B19...10 (Min Voltage) set to 5V
    - iv. B9...0 (Opr Current) set to 5A
  - d. PDO 4: Battery
    - i. B31...30 (Battery Supply) set to 01b.
    - ii. B29...20 (Max Voltage) set to 20V.
    - iii. B19...10 (Min Voltage) set to 5V.
    - iv. B9...0 (Opr. Power) set to 100W
  - e. If in PD3:
    - i. APDO1 AVS: 9 to 15V 5A and 15 to 20V 5A
      - 1. B31...30 (APDO) set to 11b.
      - 2. B29...28 (SPR AVS) set to 10b
      - 3. B27...20 (Reserved) set to zero
      - 4. B19...10 (Maximum Current 20V) set to 5A.
    - ii. APDO2 PPS: 5 to 21V 5A

1. B31...30 (APDO) set to 11b.
  2. B29...28 (SPR PPS) set to 00b.
  3. B27...25 (Reserved) set to 0.
  4. B24...17 (Min Voltage) set to 21V.
  5. B16 (Reserved) set to 0.
  6. B15...8 (Min Voltage) set to 5V.
  7. B7 (Reserved) set to 0.
  8. N6...0 (Max Current) set to 5A.
- vii. It also responds to any *Get\_Sink\_Cap\_Extended* Message with a Sink\_Capabilities\_Extended Message using the values defined in
- viii.
- ix. **Table 17** Sink\_Capabilities\_Extended Message
  - x. If Source Capabilities are received with less than the VIF specified Source Capabilities PDOs and 2 seconds have not elapsed after *PS\_RDY*, then the Tester goes back and repeats the *Request* in step 2.c.i with the Mismatch bit set.
  - xi. If the Source Capabilities matching those in the VIF have been received by the Tester and 2 seconds have not elapsed after *PS\_RDY*, repeat steps 2.a through 2.b.
  - xii. Within 2 seconds the Source Capabilities matching those in the VIF should have been received by the Tester, else the check fails, and the test stops here. [TEST.PD.PS.SRC.4#7]

Table 17 Sink\_Capabilities\_Extended Message with Sink Maximum PDP = 100W

Field	Description
VID	0x1A0A
PID	0x0302
XID	0
FW Version	0
HW Version	0
SKEDB Version	1
Load Step	0
Sink Load Characteristics	0
Compliance	0
Touch Temp	0

Field	Description
Battery Info	0
Sink Modes	Bit 0: 1 Bit 1: 1 Bit 2: 0 Bit 3: 0 Bit 4: 0 Bit 5: 1 Bit 6..7: 00
Sink Minimum PDP	Copy from VIF PD_Power_As_Source_Guaranteed, maximum of 100
Sink Operational PDP	Copy from VIF PD_Power_As_Source (Maximum PDP), maximum of 100
Sink Maximum PDP	100W
EPR Sink Minimum PDP	0
EPR Sink Operational PDP	0
EPR Sink Maximum PDP	0

#### *TEST.PD.PS.SRC.5 Source Hard Reset Test*

**Description:** As a Sink only, the Tester verifies that the Source UUT responds correctly to a Hard Reset from the Highest Voltage available.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

	Consumer Only	Provider Only, P/C, C/P, DRP
<i>Rev2Src</i>		✓
<i>Rev3ChkdSrc</i>		✓

**Test Requirements:**

**Test Procedures:**

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Consumer/Provide, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
- The Tester sends a *Hard Reset* Signal.
- The Tester makes the following checks:

- a. V<sub>BUS</sub> stays within present valid voltage range for *tPSHardReset* min after last bit of *Hard Reset* Signal. [TEST.PD.PS.SRC.5#1]
- b. V<sub>BUS</sub> reaches *vSafe0V* max within *tSafe0v* max of *tPSHardReset* max. [TEST.PD.PS.SRC.5#2]
- c. V<sub>BUS</sub> rises above *vSafe0V* max after a delay of between *tSrcRecover* min and *tSrcRecover* max after reaching *vSafe0V*. [TEST.PD.PS.SRC.5#3]
- d. V<sub>BUS</sub> reaches *vSafe5V* within *tSrcTurnOn* max after rising above *vSafe0v* max [TEST.PD.PS.SRC.5#4]
- e. The UUT starts sending a Source Capabilities message within *tFirstSourceCap* max of V<sub>BUS</sub> reaching *vSafe5v* min. [TEST.PD.PS.SRC.5#5]
4. The Tester disconnects from the UUT and waits for at least 1s.
5. If no Fixed PDO greater than 5V is available from the UUT, the test ends here.
6. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
7. The Tester makes a Request for the highest Fixed PDO available and checks that the contract is made. The Tester does not draw any current. [TEST.PD.PS.SRC.5#6]
8. The Tester sends a *Hard Reset* Signal.
9. The Tester makes the following checks:
  - a. V<sub>BUS</sub> stays within present valid voltage range for *tPSHardReset* min after last bit of *Hard Reset* Signal. [TEST.PD.PS.SRC.5#7]
  - b. V<sub>BUS</sub> reaches *vSafe5V* max within *tSafe5v* max of *tPSHardReset* max. [TEST.PD.PS.SRC.5#8]
  - c. V<sub>BUS</sub> reaches *vSafe0V* max within *tSafe0v* max of *tPSHardReset* max. [TEST.PD.PS.SRC.5#9]
  - d. V<sub>BUS</sub> rises above *vSafe0V* max after a delay of between *tSrcRecover* min and *tSrcRecover* max after reaching *vSafe0V*. [TEST.PD.PS.SRC.5#10]
  - e. V<sub>BUS</sub> reaches *vSafe5V* within *tSrcTurnOn* max after rising above *vSafe0v* max [TEST.PD.PS.SRC.5#11]
  - f. The UUT starts sending a Source Capabilities message within *tFirstSourceCap* max of V<sub>BUS</sub> reaching *vSafe5v* min. [TEST.PD.PS.SRC.5#12]
10. The Tester disconnects from the UUT.

## 6.2 Test Procedures for Sink Capable UUT

### 6.2.1 PD2 and PD3 Modes

The tests in this subsection are run firstly in PD2 mode and secondly in PD3 mode.

The tests in this subsection are applicable only to UUT with VIF field PD\_Port\_Type set to Consumer Only, Consumer/Provider and DRP.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (ie, a Provider/Consumer or Consumer/Provider) The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6

- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### *TEST.PD.PS.SNK.1 PDO Transitions As Sink*

**Description:** The Tester verifies that the UUT responds correctly to a new *Source\_Capabilities* Messages.

#### Test Specific Tester Behavior:

#### Test Conditions:

	Consumer Only, C/P, DRP	Provider Only, P/C
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. The Tester runs a bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *Get\_Sink\_Cap* Message to the UUT.
3. The check fails if the UUT does not send *Sink\_Capabilities* Message.  
[TEST.PD.PS.SNK.1#1]
4. In PD3 mode, the Testser sends a *Get\_Sink\_Cap\_Extended* Message to the UUT, the check fails if the UUT does not send a *Sink\_Capabilities\_Extended* Message.  
[TEST.PD.PS.SNK.1#19]
5. The Tester sends a new *Source\_Capabilities* Message emulating a 240W source PPS for PD3 or 100W source for PD2 as per power rules.
  - a. The first PDO is defined as:
    - i. B29 (Dual-Role Power) set to 1b.
    - ii. B28 (USB Suspend Supported) set to 0b.
    - iii. B27 (Unconstrained Power) set to 0b.
    - iv. B26 (USB Communications Capable) set to 0b.
    - v. B25 (Dual-Role Data) set to 0b.
    - vi. B24 (PD3, Unchunked Extended Message Supported) set to 0b.
    - vii. B23 (EPR Mode Capable) to 1b when in PD3, to 0b in PD2
    - viii. B22 (Reserved) set to 0b.
    - ix. B21...20 (Peak Current) set to 00b.
    - x. B19...10 (Voltage) set to 5V.
    - xi. B9...0 (Maximum Current) set to 3000mA.
  - b. Other PDOs fit with power rules:
    - i. PDO2: 9V 3A
    - ii. PDO3: 15V 3A
    - iii. PDO4: 20V 5A
    - iv. if in PD3:
      1. APDO1 AVS: 9 to 15V 3A and 15 to 20V 5A
      2. APDO2 PPS: 5 to 21V 5A
6. The check fails if the UUT does not respond with a *Request* Message.  
[TEST.PD.PS.SNK.1#2]



7. The Tester checks if requested PDO is within the range of VIF Sink Capabilities Extended PDP. [TEST.PD.PS.SNK.1#12]
  - a. For the Fixed or Variable *Request* Message, the check fails if  $B19...10$  (Operating Current)\*Voltage of the PDO in the corresponding Object Position is greater than VIF field Snk\_SPR\_Maximum PDP
  - b. For the Battery *Request* Message, the check fails if either of the following is true  $B19...10$ (Operating Power) is greater than VIF field Snk\_SPR\_Maximum PDP
  - c. For the PPS *Request* Message, the check fails if either of the following is true:
    - i. Snk\_PPS\_Charging\_Supported=N and PPS Request is sent
    - ii.  $B20...9$  (Output Voltage)\* $B6...0$ (Operating Current) is greater than VIF field Snk\_SPR\_Maximum PDP
  - d. For the SPR\_AVs, the check fails if either of the following is true:
    - i. Snk\_AVs\_Supported = N and SPR\_AVs *Request* is sent
    - ii.  $B20...9$  (Output Voltage)\* $B6...0$ (Operating Current) is greater than VIF field Snk\_SPR\_Maximum PDP.
8. The Tester intentionally sends a *Wait* Message. The test fails if another *Request* Message is received within min of *tSinkRequest*. This is measured when the EOP of the *GoodCRC* Message the *Wait* Message has been received. [TEST.PD.PS.SNK.1#17]
9. If the UUT does not resend the *Request* Message within *tCtsInfinite* second, then the Tester resends the *Source\_Capabilities* as per step 4. The test fails if the *Request* Message is not resent by the UUT or if the UUT does not respond with the *Request* Message to *Source\_Capabilities* Message resent by the Tester. [TEST.PD.PS.SNK.1#18]
10. The Tester sends an *Accept* Message.
11. If a new PDO has been requested and the transition involves increasing the voltage:
  - a. If the transition is to a Fixed Supply PDO, or to an AVS APDO with voltage change greater than *vSmallStep*.
    - i. After a total of *tSnkStdby* from the EOP of the *GoodCRC* Message acknowledging the *Accept* Message, check that current drawn by the UUT does not exceed *iSnkStdby* max. [TEST.PD.PS.SNK.1#3]
    - ii. Wait a nominal *tSrcTransition* from the *GoodCRC* Message and then change  $V_{BUS}$  voltage to the new requested value. For Fixed Supply PDO transitions, the AVS APDO transitions, the Tester shall complete the voltage change before *tAvsSrcTransLarge* expires.
    - iii. As soon as the voltage on  $V_{BUS}$  has stopped changing, check that current drawn by the UUT does not exceed *iSnkStdby* max until *PS\_RDY* is sent. [TEST.PD.PS.SNK.1#4]
  - b. If the transition is to an AVS APDO with a voltage change less than or equal to *vSmallStep*:
    - i. The Tester shall immediately begin changing  $V_{BUS}$  voltage after the *GoodCRC* Message acknowledging the *Accept* Message. The tester shall complete the voltage transition and send *PS\_RDY* before *tSrcTransSmall* expires.
    - ii. If the newly negotiated current is less than the previously contracted current, after *tSnkNewPower* from the EOP of *GoodCRC* acknowledging the *Accept* Message, check that the current drawn by the UUT does not exceed the newly negotiated current( $I_{NEW}$ ) . [TEST.PD.PS.SNK.1#20]

- iii. If the newly negotiated current is unchanged or increased, check that the current drawn by the UUT does not exceed the previously contracted current during the transition after the EOP of **GoodCRC** acknowledging the **Accept** Message until the **PS\_RDY** is sent by the Tester.  
[TEST.PD.PS.SNK.1#21]
  - c. If the transition is to a PPS APDO while in CV mode:
    - i. The Tester shall immediately begin changing  $V_{BUS}$  voltage after the **GoodCRC** Message acknowledging the **Accept** message. The Tester shall complete the voltage change and send **PS\_RDY** before  $t_{PpsSrcTransSmall}$  max for voltage changes less than or equal to  $v_{PpsSmallStep}$ , or  $t_{PpsSrcTransLarge}$  max for voltage changes greater than  $v_{PpsSmallStep}$ .
    - ii. Check that the current drawn by the UUT does not exceed the max contracted current during and after the transition.  
[TEST.PD.PS.SNK.1#22]
12. If a new PDO with no change in the voltage or no new PDO is requested, and the transition involves unchanged or decreased voltage, or no new PDO is requested:
  - a. If the newly negotiated current is **unchanged or increased**: after  $t_{SrcTransReq}$  max time period (before **PS\_RDY** is sent) check that the current drawn from  $V_{BUS}$  does not exceed the previously contracted current. [TEST.PD.PS.SNK.1#5]
  - b. If the newly negotiated current is **decreased**: after  $t_{SnkNewPower}$  from the EOP of **GoodCRC** acknowledging the **Accept** Message, check that the current drawn by the UUT does not exceed the newly negotiated current ( $I_{NEW}$ )  
[TEST.PD.PS.SNK.1#23]
13. The Tester sends a **PS\_RDY** Message to the UUT.
14. The Tester checks that the current drawn by the UUT does not exceed the level specified in the current PDO over the next 5 seconds. [TEST.PD.PS.SNK.1#6]
15. If VIF field **EPR\_Supported\_As\_Snk** is Yes and the UUT sends the **EPR\_Mode** Message with action field set to 1(Enter), the Tester sends **EPR\_Mode** Message with the Action field set to 4(Enter Failed) and the Data field set to 4(Source unable to enter EPR Mode). The check fails if the UUT does not respond with SoftReset Message  
[TEST.PD.PS.SNK.1#16]
16. If VIF field **EPR\_Supported\_As\_Snk** is No and the UUT sends the **EPR\_Mode** Message with the Action field set to 1(Enter), the test fails. [TEST.PD.PS.SNK.1#15]

#### TEST.PD.PS.SNK.2 Initial Sink PDO Transitions Post PR Swap

**Description:** The Tester verifies that the new Source UUT performs PDO transitions correctly.

##### Test Specific Tester Behavior:

- The Tester is a Dual-Role Power, initially a Source.
- The Tester sends **Sink\_Capabilities** setting B27 (Unconstrained Power) to 0b COMMON.PROC.PD.3

##### Test Conditions:

	Consumer Only, C/P, DRP	Provider Only, P/C
<b>Rev2Snk</b>	✓	

Test Procedures:

1. The Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. After this point the tester should *Reject* any *PR\_Swap* request.
3. The Tester sends a *PR\_Swap* Message.
4. The Tester checks the response: [TEST.PD.PS.SNK.2#1]
  - a. The check fails if the UUT responds with a *Reject* Message or *Not\_Supported*, and the VIF field *Accepts\_PR\_Swap\_As\_Snk* is “Yes”.
  - b. The test ends here if the UUT responds with a *Reject* Message or *Not\_Supported*.
  - c. If the UUT responds with a *Wait* Message, the Tester resends *PR\_Swap* every *tPRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. Once the Tester has repeated for *nCtsWaitCount* times, the check fails, and the test ends here.
  - d. If the UUT responds with an *Accept* Message, the Tester checks that the VIF field *PD\_Port\_Type* is set to either “Consumer/Provider” or “DRP”.
  - e. If the UUT responds with an *Accept* Message, the Tester checks that the VIF field *Accepts\_PR\_Swap\_As\_Snk* is “Yes”.
5. At *tSrcTransition* max (this delay is from the last bit of *GoodCRC* sent in response to the *Accept*), the Tester drives the  $V_{BUS}$  voltage to *vSafe0V* within *tSrcSwapStdby* max.
6. The Tester sends a *PS\_RDY* at the deadline limit of *tPSSourceOff* min, the delay is from the time the last bit of the EOP of the *GoodCRC* corresponding to the *Accept* Message.
7. The Tester checks that the UUT sends *PS\_RDY* only after the UUT has applied *vSafe5V*. [TEST.PD.PS.SNK.2#2]
8. The Tester checks that the *PS\_RDY* from the UUT is within *tNewSrc* max, this delay is from the *GoodCRC* sent by the UUT in response to the *PS\_RDY* sent by the Tester. [TEST.PD.PS.SNK.2#3]
9. Once the Tester receives *Source\_Capabilities* Message, the Tester establishes a PD contract using common procedure COMMON.PROC.PD.11.
10. After this point the Tester shall *Reject* any *PR\_Swap* request received.
11. The Tester repeats the following steps using the PDO as provided in the **PDO Sequence Table** (see **Table 14**), based on the Source PDOs specified in the VIF:
  - a. The Tester defines the following *Request* Message:
    - i. B27 (Reserved) = 0b
    - ii. B26 (Capability Mismatch) = 0b
    - iii. B25 (USB Communication Capable) = 0b
    - iv. B24 (No USB Suspend) = 1b
    - v. For Fixed PDO:
 

B19...10 (Operating Current / Power) = 100mA / 0.5W

B9...0 (Maximum Operating Current / Power) = Maximum Current / Power in the Source PDO.
    - vi. For PPS PDO:
 

B20...9 (Output Voltage) = Maximum Voltage offered in the last Fixed PDO

B6...0 (Operating Current) = 1A
    - vii. For SPR AVS APDO:

B20...9 (Output Voltage) = Maximum Voltage offered in Source APDO

B6...0 (Operating Current) = 1A

- b. If VIF parameter Has\_Invariant\_PDOs is set to Y, or the current Source Capabilities match those in the VIF:
  - i. The Tester sends *Request* Message defined above.
  - ii. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
  - iii. If *Accept* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SNK.2#11]
  - iv. If the first Preamble bit of *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [TEST.PD.PS.SNK.2#12]
- c. Otherwise, If VIF parameter Has\_Invariant\_PDOs is set to N, and the current Source Capabilities do not match those in the VIF:
  - i. The Tester repeats sending the previous *Request* Message (for what is currently contracted) but with the Mismatch bit set (this prevents interfering with power test procedure, by not changing the current contract)
  - ii. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
  - iii. If an *Accept* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SNK.2#11]
  - iv. If the first Preamble bit of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [TEST.PD.PS.SNK.2#12]
  - v. The Tester responds to any *Get\_Sink\_Cap* Message with *Sink\_Capabilities* Message using the values found in the UUT VIF Source PDO values. It also responds to any *Get\_Sink\_Cap\_Extended* Message using the values defined in **Table 16**.
  - vi. If Source Capabilities are received with less than the Sink Capabilities (same as VIF Source Capabilities) and *tCapabilitiesMismatchResponse* has not elapsed, then the Tester goes back and repeats the *Request* Message in step I with the Mismatch bit set.
  - vii. Within *tCapabilitiesMismatchResponse*,
    - 1. If the Source Capabilities matching those in the VIF have been received by the Tester, the Tester sends the *Request* Message defined above with B19..10 (Operating Current) replaced with

- Maximum Current in Source PDO. The Tester passes the check and proceeds to the next step. [TEST.PD.PS.SNK.2#13]
2. Else, run **Check 1** and **Check 2** from COMMON.PROC.PD.18 [TEST.PD.PS.SNK.2#14]
- viii. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
  - ix. If *Accept* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SNK.2#15]
- d. The Tester performs the following checks:
 

Decrease the current drawn by the Tester to less than *iSnkStdby* within *tSnkStdby* min of the last bit of the *GoodCRC* sent in response to the *Accept* Message. The Tester performs the following checks:

    1. The Tester checks the  $V_{BUS}$  is within *vSrcNew* or *vPpsNew* of the starting voltage within *tSrcTransition* min after the end of the *GoodCRC* sent in response to the *Accept* Message. [TEST.PD.PS.SNK.2#6]
    2. The Tester checks that the  $V_{BUS}$  transition meets *vSrcSlewPos* and *vSrcSlewNeg*. [TEST.PD.PS.SNK.2#7] (see **Appendix E. Slew Rate Measurements.** )
    3. During the voltage transition, the Tester checks that the  $V_{BUS}$  voltage remains within *vSrcValid* limits upon crossing into the *vSrcValid* limits until *tSrcSettle* max. [TEST.PD.PS.SNK.2#8]
    4. The Tester checks that the  $V_{BUS}$  voltage is within *vSrcNew* or *vPpsNew* limits between *tSrcSettle* max and 100ms after *tSrcSettle* max. [TEST.PD.PS.SNK.2#9]
    5. The Tester checks that the UUT does not send *PS\_RDY* before the  $V_{BUS}$  is within *vSrcNew* or *vPpsNew*. [TEST.PD.PS.SNK.2#10]

### TEST.PD.PS.SNK.3 Multiple Request Load Test Post PR Swap

**Description:** The Tester verifies that the new Source UUT responds correctly to various *Request* Messages.

#### Test Specific Tester Behavior:

- The Tester is a Dual-Role Power, initially a Source.
- The Tester sends *Sink\_Capabilities* setting B27 (Unconstrained Power) to 0b COMMON.PROC.PD.3

#### Test Conditions:

	C/P, DRP	Consumer Only, Provider Only, P/C
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

Test Requirements: If Captive\_Cable = YES, *vSrcNew* min, *vAvsNew* min or *vPpsNew* min limit shall be (*vSrcNew* min or *vAvsNew* min) - (0.75\*I/X) or (vPpsNew min) - (0.75\*I/X), where I is the actual current being drawn, and X=3 if the sourced PDO contract is within the range (0, 3A], or X=5 if the sourced PDO contract is within the range (3A, 5A)

Test Procedures:

1. The Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends a *PR\_Swap* Message.
3. The Tester checks the response: [TEST.PD.PS.SNK.3#1]
  - a. The check fails if the UUT responds with a *Reject* Message or *Not\_Supported*, and the VIF field Accepts\_PR\_Swap\_As\_Snk is “Yes”.
  - b. The test ends here if the UUT responds with a *Reject* Message or *Not\_Supported*.
  - c. If the UUT responds with a *Wait* Message, the Tester resends *PR\_Swap* every *tPRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. Once the Tester has repeated for *nCtsWaitCount* times, the check fails, and the test ends here.
  - d. If the UUT responds with an *Accept* Message, the Tester checks that the VIF field PD\_Port\_Type is set to either “Consumer/Provider” or “DRP”.
  - e. If the UUT responds with an *Accept* Message, the Tester checks that the VIF field Accepts\_PR\_Swap\_As\_Snk is “Yes”.
4. At *tSinkTransition* max (this delay is from the last bit of *GoodCRC* sent in response to the *Accept*), the Tester drives the V<sub>BUS</sub> voltage to *vSafe0V* within *tSrcSwapStdby* max.
5. The Tester sends a *PS\_RDY* at the deadline limit of *tPSSourceOff* min, the delay is from the time the last bit of the EOP of the *GoodCRC* corresponding to the *Accept* Message.
6. The Tester checks that the UUT sends *PS\_RDY* only after the UUT has applied *vSafe5V*. [TEST.PD.PS.SNK.3#2]
7. The Tester checks that the *PS\_RDY* from the UUT is within *tNewSrc* max, this delay is from the *GoodCRC* sent by the UUT in response to the *PS\_RDY* sent by the Tester. [TEST.PD.PS.SNK.3#3]
8. Once the Tester receives *Source\_Capabilities* Message, the Tester establishes a PD contract using common procedure COMMON.PROC.PD.11.
9. After this point the Tester shall reject any *PR\_Swap* request received.
10. The Tester repeats the following steps for each of the PDOs (specified in the VIF), starting with the first PDO:
  - a. If the next Request will be for a PPS transition involving a current decrease (but no PDO change), the Tester reduces its current draw to the new level required, before sending the Request. (This avoids the current limit in the UUT shutting off V<sub>BUS</sub>.) The Tester draws the requested Operating Current / Power with a current transition slew rate of 100mA per μs. The Tester monitors V<sub>BUS</sub> voltage and the check fails if V<sub>BUS</sub> voltage is not within the limits of *PpsNew*. [TEST.PD.PS.SNK.3#4]
  - b. The Tester defines *Request* Message as following:
    - i. B27 (Reserved) = 0b
    - ii. B26 (Capability Mismatch) = 0b
    - iii. B25 (USB Communication Capable) = 0b



- iv. B24 (No USB Suspend) = 1b
- v. For Fixed, Battery or Variable PDO:
  - 1. B19...10 (Operating Current / Power):
    - a. For first Request: 0mA/0mW
    - b. For subsequent Requests: As defined in Step 10.f.i
  - 2. B9...0 (Maximum Operating Current / Power) = Maximum Current / Power in the Source PDO.
- vi. For PPS:
  - 1. B20...9 (Output Voltage) = Maximum Voltage offered in Source APDO
  - 2. B6...0 (Operating Current):
    - a. For first Request: 1A
    - b. For subsequent Requests: As defined in Step 10.f.ii.
- vii. For SPR AVS:
  - 1. B20...9 (Output Voltage) = Iterate through Min, Max and Mid Voltage offered in Source Fixed PDO
  - 2. B6...0 (Operating Current):
    - a. For first Request: 0mA/0mW
    - b. For subsequent Requests: As defined in Step 10.f.i.

The initial Request is for PDO #1 (5V Fixed) at 0mA. Further requests are defined in step 10f.

- c. If VIF parameter Has\_Invariant\_PDOs is set to Y, or the current Source Capabilities match those in the VIF:
  - i. The Tester sends *Request* Message defined above.
  - ii. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
  - iii. If *Accept* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SNK.3#12]
  - iv. If a *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [TEST.PD.PS.SNK.3#13]
- d. Otherwise, If VIF parameter Has\_Invariant\_PDOs is set to N, and the current Source Capabilities do not match those in the VIF:
  - i. The Tester repeats sending the previous *Request* Message (for what is currently contracted) but with the Mismatch bit set (this prevents interfering with power test procedure, by not changing the current contract)
  - ii. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
  - iii. If an *Accept* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SNK.3#12]



- iv. If the first Preamble bit of *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to an *Accept* Message, the check fails, and the test stops here. [TEST.PD.PS.SNK.3#13]
- v. The Tester responds to any *Get\_Sink\_Cap* Message with *Sink\_Capabilities* Message using the values found in the UUT VIF Source PDO values. It also responds to any *Get\_Sink\_Cap\_Extended* Message using the values defined in the **Table 16**.
- vi. If Source Capabilities are received with less than the Sink Capabilities (same as VIF Source Capabilities) and *tCapabilitiesMismatchResponse* has not elapsed, then the Tester goes back and repeats the *Request* Message in step I with the Mismatch bit set.
- vii. Within *tCapabilitiesMismatchResponse*
  - 1. the Source Capabilities matching those in the VIF have been received by the Tester, it sends the *Request* Message defined above. The Tester passes the check and proceeds to the next step. [TEST.PD.PS.SNK.3#14]
  - 2. Else, run **Check 1** and **Check 2** from COMMON.PROC.PD.18 [TEST.PD.PS.SNK.3#15]
- viii. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of *nCtsWaitCount* times or until *Accept* is received.
- ix. If *Accept* Message is not received, the check fails, and the test stops here. [TEST.PD.PS.SNK.3#16]
- e. The Tester performs the following checks:
  - i. If the transition involves a current decrease (but no PDO change) [TEST.PD.PS.SNK.3#4]
    - 1. For Fixed, Battery or Variable PDOs:  
The Tester decreases current to the new value within *tSnkNewPower* min of the last bit of the *GoodCRC* sent in response to the *Accept* Message. The Tester draws the requested Operating Current / Power with a current transition slew rate of 100mA per  $\mu$ s. The Tester monitors  $V_{BUS}$  voltage and the check fails if  $V_{BUS}$  voltage is not within the limits of *vSrcNew*.
    - 2. For PPS:  
The current was already reduced in step 10a.
  - ii. If the transition involves a current increase (but no PDO change), the Tester increase current to the new value after the last bit of the *GoodCRC* sent in response to the *PS\_RDY* Message. The Tester draws the requested Operating Current / Power with a current transition slew rate of 100mA per  $\mu$ s. The Tester monitors  $V_{BUS}$  voltage and the check fails if  $V_{BUS}$  voltage is not within the limits of *vSrcNew* or *vPpsNew*. [TEST.PD.PS.SNK.3#5]

- iii. If the transition involves a change of PDO with a voltage increase, decrease the current drawn by the Tester to less than *iSnkStdby* within *tSnkStdby* min of the last bit of the *GoodCRC* sent in response to the *Accept* Message. If the transition involves a change of PDO with decreasing voltage and current, the Tester decreases its current draw to the requested current or power for Fixed, Battery, Variable and SPR\_AVS PDOs, 80% of the requested current for PPS PDO within *tSrcTransition* min after the end of the *GoodCRC* is sent in response to the *Accept* Message. The Tester performs the following checks:
  1. The Tester checks that  $V_{BUS}$  remains within *vSrcNew*, *vAvsNew* or *vPpsNew* of the starting voltage within *tSrcTransition* min after the end of the *GoodCRC* sent in response to the *Accept* Message. [TEST.PD.PS.SNK.3#6]
  2. The Tester checks that the  $V_{BUS}$  transition meets *vSrcSlewPos* and *vSrcSlewNeg*. [TEST.PD.PS.SNK.3#7] (see **Appendix E. Slew Rate Measurements.** )
  3. During the voltage transition, the Tester checks that the  $V_{BUS}$  voltage remains within *vSrcValid* limits from the time of crossing into the *vSrcValid* limits until *tSrcSettle* max. [TEST.PD.PS.SNK.3#8]
  4. The Tester checks that the  $V_{BUS}$  voltage remains within *vSrcNew*, *vAvsNew* or *vPpsNew* limits between *tSrcSettle* max and 100ms after *tSrcSettle* max. [TEST.PD.PS.SNK.3#9]
  5. The Tester checks that the UUT does not send *PS\_RDY*, if the transition involves a current increase, before the  $V_{BUS}$  is within *vSrcNew*, *vAvsNew* or *vPpsNew*. [TEST.PD.PS.SNK.3#10]
  6. After having sent *GoodCRC* in response to *PS\_RDY*, if the transition involves a current increase, the Tester increases its current draw to:
    - a. For Fixed, Battery and Variable and SPR\_AVS PDOs: the current or power requested.
    - b. For PPS PDO: 80% of the current requested (except that for the last step 0mA will be drawn)
  7. The Tester checks that the  $V_{BUS}$  voltage remains within *vSrcNew*, *vAvsNew* or *vPpsNew* limits for 100ms after receiving *PS\_RDY*. [TEST.PD.PS.SNK.3#11]
- f. The Tester repeats Step-10a through Step-10d, replacing the appropriate fields with the next value from the following list. The Tester sends 9 separate *Request* Messages for each PDO (7 separate *Request* Messages for PPS, 9 separate *Request* Messages for each SPR\_AVS min, max and mid voltage iteration):
  - i. For Fixed, Battery and Variable PDOs, replace B19...10 (Operating Current / Power):
    1. 0mA / 0mW [performed in Step 10.b; listed here for reference only]

2. 25% of Maximum Current / Power in the Source PDO
  3. 50% of Maximum Current / Power in the Source PDO
  4. 75% of Maximum Current / Power in the Source PDO
  5. 100% of Maximum Current / Power in the Source PDO
  6. 75% of Maximum Current / Power in the Source PDO
  7. 50% of Maximum Current / Power in the Source PDO
  8. 25% of Maximum Current / Power in the Source PDO
  9. 0mA / 0mW
- ii. For PPS APDO, replace B20...9 (Output Voltage) and B6...0 (Operating Current):
1. Maximum Voltage offered in Source APDO @ 1A (Tester will draw 0mA) [performed in Step 10.b; listed here for reference only]
  2. Maximum Voltage offered in Source APDO @ 1A. (Tester draws 80% load)
  3. Maximum Voltage offered in Source APDO @  $(1A + \text{Maximum Current offered in Source APDO})/2$ . (Tester draws 80% load)
  4. Maximum Voltage offered in Source APDO @ Maximum Current offered in Source APDO. (Tester draws 80% load)
  5. Maximum Voltage offered in Source APDO @  $(1A + \text{Maximum Current offered in Source APDO})/2$ . (Tester draws 80% load)
  6. Maximum Voltage offered in Source APDO @ 1A (Tester draws 80% load)
  7. Maximum Voltage offered in Source APDO @ 1A (Tester draws 0mA)

## 7 Extended Power Range Tests

### 7.1 Test Procedure for EPR Source Capable UUT

#### 7.1.1 PD3 Mode

The tests in this subsection are run in PD3 mode.

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (ie, a Provider/Consumer or Consumer/Provider) The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always rejects *VCONN\_Swap* COMMON.PROC.PD.6, unless mentioned otherwise in the test procedure.
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7

- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8
- The Tester sets the *EPR\_Mode\_Capable* bit to 1 in the Request and *EPR\_Request* Message, unless mentioned otherwise in the test procedure.
- If a SOP' *Discover Identity* Command is received from the UUT, the Tester replies as Passive Cable with  $V_{BUS}$  Current Handling Capability set to 10b(5A), Maximum  $V_{BUS}$  Voltage set to 11b(50V) and EPR Mode Capable field set to 1b (EPR Mode Capable), unless mentioned otherwise in the test procedure
- Configure Tester's *Sink\_Capabilities\_Extended* Message as follows:
  - B21 (EPR Sink Minimum PDP): 140
  - B22 (EPR Sink Operational PDP): 140
  - B23 (EPR Sink Maximum PDP): 140

#### TEST.PD.EPR.SRC3.1 EPR Entry Process - UUT as VCONN Source

Description: The Tester as a sink verifies the successful EPR negotiation when UUT acts as a VCONN source.

Test Specific Tester Behavior:

Test Conditions:

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓
Rev3UnChkdSrc		✓	✓

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester behaves as a Provider/Consumer, and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7. If the UUT does not accept  $V_{CONN\_Swap}$  to become  $V_{CONN}$  Source, the test passes and stops here.
2. The Tester sends the *EPR\_Get\_Source\_Cap* Message and perform below check  
[TEST.PD.EPR.SRC3.1#13]

- a. If the UUT is Provider Only and VIF field `EPR_Supported_as_SRC` is No, the tester checks that the UUT responds with the *Not\_Supported* Message.
  - b. If the UUT is DRP or C/P or P/C and VIF field `EPR_Supported_as_SNK` is No and VIF field `EPR_Supported_as_SRC` = No , the Tester checks that the UUT responds with the *Not\_Supported* Message.
  - c. If the UUT is not Provider Only and VIF field `EPR_Supported_as_SRC` is No and VIF field `EPR_Supported_as_Snk` is Yes, the Tester checks that the UUT responds the *Reject* or *Not\_Supported* Message.
  - d. If the UUT VIF field `EPR_Supported_as_SRC` is Yes, the Tester checks that the UUT responds with the *EPR\_Source\_Capabilities* Message.
- Note: *EPR\_Source\_Capabilities* should not be the maximum power the UUT is able to offer.
3. The Tester sends an *EPR\_Mode* ( Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)
  4. The Tester performs below [TEST.PD.EPR.SRC3.1#2]
    - a) If the VIF field `EPR_Supported_As_Src` is Yes, The Tester checks that the UUT responds with *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x00.
    - b) If the VIF field `EPR_Supported_As_Src` is No, The Tester checks that the UUT responds with *Not\_Supported* Message and the test stops here.
  5. Tester checks that UUT completes the EPR entry process by sending *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) and Data field set to 0x0 within *tEnterEPR*, this timing is measured from the last bit of the *GoodCRC* in response to the *EPR\_Mode* (Enter) Message to the last bit of the *EPR\_Mode* (Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.1#5] . If the check fails, the Test stops here.
  6. If the VIF field *Captive\_Cable* is No, the Tester performs below checks immediately after receiving the *EPR\_Mode* (Enter\_Succeeded) Message.
    - a. The UUT did not send a *VCONN\_Swap* Message [TEST.PD.EPR.SRC3.1#3]
    - b. The UUT did not send SOP' *Discover Identity* Command prior [TEST.PD.EPR.SRC3.1#4]

7. The Tester checks that the UUT sends *EPR\_Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* (Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.1#6]
8. The Tester checks that the UUT establishes EPR contract using common procedure COMMON.PROC.PD3.2 [TEST.PD.EPR.SRC3.1#10]
9. The Tester waits for 2 seconds. If the UUT sends *EPR\_Source\_Capabilities* Message with no EPR PDOs, the Tester checks that the UUT completes the contract negotiation using common procedure COMMON.PROC.PD3.2, otherwise, the Tester performs the following steps:
  - a. The Tester sends *EPR\_Request* Message with Object Position set to 1, Operating Current and Maximum Operating Current set to 100mA
  - b. The check fails if either *Accept*, *Reject* or *Wait* Message is not received, and the test stops here. If the *Reject* Message is received, the Tester reports the check with a warning, and the test stops here. [TEST.PD.EPR.SRC3.1#7]
  - c. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after *nCtsWaitCount* retries, and the test stops here. [TEST.PD.EPR.SRC3.1#12]
  - d. The check fails if the first Preamble bit of *PS\_RDY* Message is not received within EPR's mode *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [TEST.PD.EPR.SRC3.1#8]
10. Tester waits for 100ms for UUT to initiate the *EPR\_Mode* Message with Action field set 0x5(Exit) and Data field set to 0, else the Tester sends the *EPR\_Mode* Message with Action field set 0x5(Exit) and Data field set to 0.
11. Tester checks that the UUT sends *Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* (Exit) Message [TEST.PD.EPR.SRC3.1#9]
12. Tester checks that the UUT completes the contract negotiation using common procedure COMMON.PROC.PD.11 [TEST.PD.EPR.SRC3.1#11]

### TEST.PD.EPR.SRC3.2 EPR Entry Process - Tester as VCONN Source

Description: The Tester as a sink verifies the successful EPR negotiation when Tester acts as a VCONN source.

Test Specific Tester Behavior: Tester rejects the VCONN Swap message initiated by UUT unless mentioned in the test procedure.

#### Test Conditions:

This test is not applicable if the VIF parameter EPR\_Supported\_As\_Src is set to No or if the Captive\_Cable field in the VIF is set to Yes.

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓
Rev3UnchkdSrc		✓	✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester behaves as a Provider/Consumer, and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
2. The Tester sends a *VCONN\_Swap* message.
3. The Tester checks the *Accept* Message response from the UUT.  
[TEST.PD.EPR.SRC3.2#2]



4. The Tester sends a *PS\_RDY* Message.
5. The Tester checks that the UUT does not present VCONN at the non-CC line within *tVCONNSourceOff*. [TEST.PD.EPR.SRC3.2#3]
6. Configure the Tester to accept the *VCONN\_Swap* message and then Tester sends an *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)
7. The Tester checks that the UUT responds *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0, otherwise the check fails and stops here [TEST.PD.EPR.SRC3.2#4]
8. If the UUT sends a *VCONN\_Swap* Message within *tEPRSourceCableDiscovery*, go to step 10
9. If the UUT has already sent SOP' *Discover Identity* Command, go to step 13.
10. The Tester checks that UUT completes the V<sub>CONN</sub> swap sequence using COMMON.PROC.PD.6
11. If a SOP' *Discover Identity* Command is received from the UUT, the Tester replies as Passive Cable with V<sub>BUS</sub> Current Handling Capability set to 10b(5A), Maximum V<sub>BUS</sub> Voltage set to 11b(50V) and EPR Mode Capable field set to 1b (EPR Mode Capable)
12. The Tester checks that UUT completes the EPR entry process by sending *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) and Data field set to 0x0 within *tEnterEPR*, this timing is measured from the last bit of the *GoodCRC* in response to the *EPR\_Mode* (Enter) Message to the last bit of the *EPR\_Mode* (Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.2#7]
13. The Tester checks the UUT has sent the SOP' *Discover Identity* Command any time prior sending the *EPR\_Mode* (Enter\_Succeeded) Message. [TEST.PD.EPR.SRC3.2#6]
14. Tester checks that the UUT sends *EPR\_Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* (Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.2#8]
15. Tester checks that the UUT establishes EPR contract using common procedure COMMON.PROC.PD3.2 [TEST.PD.EPR.SRC3.2#9]

16. Tester waits for 2 seconds, If UUT sends *EPR\_Source\_Capabilities* Message with no EPR PDOs, Tester checks that UUT completes the contract negotiation using common procedure COMMON.PROC.PD3.2, otherwise, the Tester performs below steps
- a. Tester sends *EPR\_Request* Message with Object Position set to 1, Operating Current and Maximum Operating Current set to 100mA
  - b. The check fails if an *Accept* or *Wait* Message is not received.  
[TEST.PD.EPR.SRC3.2#10]
  - c. If the UUT responds with the *Wait* Message, the Tester resends the *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until the *Accept* Message is received. The check fails if the UUT responds with the *Wait* Message after *nCtsWaitCount* retries, and the test stops here.  
[TEST.PD.EPR.SRC3.2#14]
  - d. The check fails if the first Preamble bit of *PS\_RDY* Message is not received within EPR's mode *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message.  
[TEST.PD.EPR.SRC3.2#11]
17. Tester waits for 100ms for UUT to initiate the *EPR\_Mode* Message with Action field set 0x5(Exit) and Data field set to 0, else the Tester sends the *EPR\_Mode* Message with Action field set 0x5(Exit) and Data field set to 0
18. Tester checks that the UUT sends *Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* (Exit) Message [TEST.PD.EPR.SRC3.2#12]
19. Tester checks that the UUT completes the contract negotiation using common procedure COMMON.PROC.PD.11 [TEST.PD.EPR.SRC3.2#13]

#### TEST.PD.EPR.SRC3.3 EPR Entry failed - EPR Mode Capable bit not set in RDO

Description: As a Sink, the Tester verifies that the Source UUT fails the EPR entry process when the EPR Mode Capable bit was not set in the most recent RDO.

Test Specific Tester Behavior:

Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Src* is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1. (The Tester sends a *Request* Message with EPR Mode Capable Bit, B22, set to 0 )
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7 (The Tester sends a *Request* Message with EPR Mode Capable Bit, B22, set to 0 )
2. The Tester sends an *EPR\_Mode* ( Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)
3. The Tester checks that UUT responds with *EPR\_Mode* Message with the Action field set to 0x04(Enter Failed) and Data field set to 0x03(EPR Mode Capable bit not set in RDO). [TEST.PD.EPR.SRC3.3#2]
4. Tester sends the *Soft Reset* Message and checks the UUT sends the *Accept* Message followed by the *Source\_Capabilities* Message [TEST.PD.EPR.SRC3.3#3]
5. Tester checks that the UUT establishes a SPR contract negotiation using COMMON.PROC.PD.11 [TEST.PD.EPR.SRC3.3#5]
6. Tester waits for 2 seconds and checks that the UUT does not initiate the *EPR\_Source\_Capabilities* Message [TEST.PD.EPR.SRC3.3#4]

#### TEST.PD.EPR.SRC3.4 EPR Entry failed – Tester as VCONN source

Description: The Tester as a sink, verifies the UUT fails the EPR entry process while Tester is acting as the VCONN source.

#### Test Specific Tester Behavior:

#### Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Src* is set to No or if the *Captive\_Cable* field in the VIF is set to Yes

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓

### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (set EPR mode bit in RDO to 1).
  - b. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider, the Tester behaves as a Provider/Consumer, and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
2. The Tester sends a *VCONN\_Swap* message.
3. The Tester checks the *Accept* Message response from the UUT.  
[TEST.PD.EPR.SRC3.4#2]
4. The Tester sends a *PS\_RDY* Message.
5. The Tester checks that the UUT does not present VCONN at the non-CC line within *tVCONNSourceOff*. [TEST.PD.EPR.SRC3.4#3]
6. Tester sends an *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)
7. The Tester checks that the UUT responds *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0, otherwise the check fails and stops here [TEST.PD.EPR.SRC3.4#4]
8. if the UUT sends a *VCONN\_Swap* Message [TEST.PD.EPR.SRC3.4#5] within *tEPRSourceCableDiscovery*, go to step 10.
9. If the UUT has already sent SOP' *Discover Identity* Command, the Tester passes the check, otherwise the check fails. [TEST.PD.EPR.SRC3.4#9] The test stops here.

10. The Tester checks that the UUT sends *EPR\_Mode* with Action field set to 0x4(Enter Failed) and Data field set to 0x02(Source failed to become VCONN source).  
[TEST.PD.EPR.SRC3.4#6]
11. If the EPR entry process does not exit within *tEnterEPR* (this timing is measured from the last bit of the *GoodCRC* in response to the *EPR\_Mode* (Enter) Message to the last bit of the *EPR\_Mode* (Enter\_Failed) Message, the check fails  
[TEST.PD.EPR.SRC3.4#7]

#### TEST.PD.EPR.SRC3.5 EPR Entry Failed - *EPR\_Mode* (Reserved) Message

Description: As a Sink, the Tester verifies that the Source UUT responds with *Soft Reset* to the *EPR\_Mode* Reserved message.

Test Specific Tester Behavior:

Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Src* is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (set EPR mode bit in RDO to 1).
  - b. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider, the Tester behaves as a Provider/Consumer, and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
2. The Tester sends an *EPR\_Mode* (Enter) Message with the Action field set to 0x00(Reserved) and Data field set to 140 (140W).

3. The Tester checks that UUT sends a *Soft Reset* Message within *tProtErrSoftReset* (this timing is measured from the last bit of the *GoodCRC* of *EPR\_Mode* (Enter) Message to the last bit of the *Soft Reset* Message) [TEST.PD.EPR.SRC3.5#2]

#### TEST.PD.EPR.SRC3.6 EPR Entry Failed - Cable not EPR capable

Description: As a Sink, the Tester emulates as a non EPR cable and verifies the UUT fails the EPR entry process

Test Specific Tester Behavior:

Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Src* is set to No or *Captive\_Cable* field is set to Yes.

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓

Cable field configurations:

1. Configuration 1:
  - a. VBUS Current Handling Capability set to 10b(5A),
  - b. Maximum VBUS Voltage set to 11b(50V),
  - c. EPR Mode Capable field set to 0b (EPR Mode Not Capable)
2. Configuration 2:
  - a. VBUS Current Handling Capability set to 10b(5A),
  - b. Maximum VBUS Voltage set to 00b(20V),
  - c. EPR Mode Capable field set to 1b (EPR Mode Capable)
3. Configuration 3:
  - a. VBUS Current Handling Capability set to 01b(3A),
  - b. Maximum VBUS Voltage set to 11b(50V),
  - c. EPR Mode Capable field set to 1b (EPR Mode Capable)

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.

If SOP' *Discover Identity* Command is received from the UUT during bring-up, the Tester replies as a Passive Cable with the cable field configuration listed above (using the appropriate configuration).

- b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester behaves as a Provider/Consumer, and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
2. The Tester sends an *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)
  3. The Tester checks that UUT responds *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0, otherwise the check fails and stops here [TEST.PD.EPR.SRC3.6#2]
  4. If a SOP' *Discover Identity* Command is received from the UUT, the Tester replies as a PassiveCable with the appropriate cable field configuration listed above.  
  
Else, if SOP' *Discover Identity* Command has been already received from the UUT, go to step 6.
  5. The check fails if the UUT does not send SOP' *Discover Identity* Command [TEST.PD.EPR.SRC3.6#3]
  6. The Tester checks that UUT responds *EPR\_Mode* Message with the Action field set to 0x04(Enter Failed) and Data field set to 0x01(Cable not EPR capable). [TEST.PD.EPR.SRC3.6#4]
  7. If the EPR entry process does not exit within *tEnterEPR* (this timing is measured from the last bit of the *GoodCRC* in response to the *EPR\_Mode* (Enter) Message to the last bit of the *EPR\_Mode* (Enter\_Failed) Message, the check fails [TEST.PD.EPR.SRC3.6#5]
  8. The Tester moves to disconnect state.
  9. Repeat the test with the next cable field configuration listed above.

#### TEST.PD.EPR.SRC3.7 EPR Entry Failed - Interrupted by EPR\_Get\_Sink\_Cap message

Description: As a Sink, the Tester verifies that the Source UUT responds with *Soft Reset* when EPR Mode entry process Interrupted by *EPR\_Get\_Sink\_Cap* Message

Test Specific Tester Behavior:

Test Conditions:



This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Src* is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓

#### Test Procedures:

1. There are 3 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (set EPR mode bit in RDO to 1).
  - b. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider, the Tester behaves as a Provider/Consumer, and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
2. The Tester sends an *EPR\_Mode* ( Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)
3. The Tester checks that UUT responds *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0, otherwise the check fails and stops here [TEST.PD.EPR.SRC3.7#2]
4. Tester sends an *EPR\_Get\_Sink\_Cap* Message between 25 - 26us from the last bit of *GoodCRC* of the *EPR\_Mode* ( Enter Acknowledged) Message.
5. The check fails if the UUT responds with *EPR\_Sink\_Cap* or any message other than *Soft Reset* Message within *tProtErrSoftReset* (this timing is measured from the last bit of the *GoodCRC* of *EPR\_Get\_Sink\_Cap* Message to the last bit of the *Soft Reset* Message) [TEST.PD.EPR.SRC3.7#3]

#### TEST.PD.EPR.SRC3.8 EPR mode - *Request* Message response

Description: As a Sink, the Tester verifies that the Source UUT responds with *Hard Reset* to the *Request* Message received in the *EPR\_Mode*.

#### Test Specific Tester Behavior:

## Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Src* is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Provider Only, Provider/Consumer or *DRP*, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source *COMMON.PROC.BU.1* (set *EPR* mode bit in *RDO* to 1).
  - b. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider, the Tester behaves as a Provider/Consumer, and it runs bring-up procedure with the UUT as a Source *COMMON.PROC.BU.7*
2. The Tester sends an *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)
3. The Tester checks that UUT responds *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0, otherwise the check fails and stops here [TEST.PD.EPR.SRC3.8#2]
4. If VIF field *Captive\_Cable* = Yes, skip to step 8, else Tester checks that UUT does not sends a *VCONN\_Swap* message [TEST.PD.EPR.SRC3.8#3]
5. If a SOP' *Discover Identity* Command is received from the UUT, the Tester replies as Passive Cable with *V<sub>BUS</sub>* Current Handling Capability set to 10b(5A), Maximum *V<sub>BUS</sub>* Voltage set to 11b(50V) and *EPR Mode Capable* field set to 1b (*EPR Mode Capable*)
6. The Tester checks if the UUT has sent SOP' *Discover Identity* Command any time prior sending the *EPR\_Mode*(Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.8#4]
7. Tester checks that UUT completes the *EPR* entry process by sending *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) and Data field set to 0x0 within *EnterEPR*, this timing is measured from the last bit of the *GoodCRC* in response to the *EPR\_Mode* (Enter) Message to the last bit of the *EPR\_Mode* (Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.8#5]

8. Tester checks that the UUT sends *EPR\_Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* (Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.8#6]
9. Tester responds with a *Request* Message with Object Position set to 1 and other fields set to values mentioned in bring-up procedure BU1.
10. Tester checks that the UUT sends a *Hard Reset* Signaling within *tProtErrHardReset*(this timing is measured from the last bit of the *GoodCRC* of *Request* Message to the last bit of the *Hard Reset* Signal)[TEST.PD.EPR.SRC3.8#7]

#### TEST.PD.EPR.SRC3.9 EPR mode - EPR\_Get\_Source\_Cap Message

Description: As a Sink, the Tester verifies that the Source UUT responds correctly for *EPR\_Get\_Source\_Cap* Message in the EPR\_Mode.

Test Specific Tester Behavior:

Test Conditions:

This test is not applicable if the VIF parameter EPR\_Supported\_As\_Src is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓
Rev3UnchkdSrc		✓	✓

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (set EPR mode bit in RDO to 1).
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester behaves as a Provider/Consumer, and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
2. The Tester sends an *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)

3. The Tester checks that UUT responds *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0, otherwise the check fails and stops here [TEST.PD.EPR.SRC3.9#2]
4. If VIF field Captive\_Cable is Yes, skip to step 10, else Tester checks that UUT does not send a *VCONN\_Swap* message [TEST.PD.EPR.SRC3.9#3]
5. If a SOP' *Discover Identity* Command is received from the UUT, the Tester replies as Passive Cable with V<sub>BUS</sub> Current Handling Capability set to 10b(5A), Maximum V<sub>BUS</sub> Voltage set to 11b(50V) and EPR Mode Capable field set to 1b (EPR Mode Capable)
6. The Tester checks if the UUT has sent SOP' *Discover Identity* Command any time prior sending the *EPR\_Mode*(Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.9#4]
7. The Tester checks that UUT completes the EPR entry process by sending *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) and Data field set to 0x0 within *tEnterEPR*, this timing is measured from the last bit of the *GoodCRC* in response to the *EPR\_Mode* (Enter) Message to the last bit of the *EPR\_Mode* (Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.9#5]
8. The Tester checks that the UUT sends *EPR\_Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* ( Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.9#6]
9. Tester checks that the UUT establishes EPR contract using common procedure COMMON.PROC.PD3.2 [TEST.PD.EPR.SRC3.9#11]
10. Tester sends the *EPR\_Get\_Source\_Cap* Message.
11. The check fails if the UUT does not send the *EPR\_Source\_Capabilities* Message [TEST.PD.EPR.SRC3.9#7]
12. Tester checks that the UUT establishes EPR contract using common procedure COMMON.PROC.PD3.2[TEST.PD.EPR.SRC3.9#12]
13. Tester waits for 2 seconds, If UUT sends *EPR\_Source\_Capabilities* Message with no EPR PDOs, Tester checks that UUT completes the contract negotiation using common procedure COMMON.PROC.PD3.2, otherwise, the Tester performs below steps
  - a. Tester sends *EPR\_Request* Message with Object Position set to 1, Operating Current and Maximum Operating Current set to 100mA.

- b. The check fails if either *Accept*, *Reject* or *Wait* Message is not received, and the test stops here. If the *Reject* Message is received, the Tester reports the check with a warning, and the test stops here. [TEST.PD.EPR.SRC3.9#8]
  - c. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after *nCtsWaitCount* retries, and the test stops here [TEST.PD.EPR.SRC3.9#14]
  - d. The check fails if the first Preamble bit of *PS\_RDY* Message is not received within EPR's mode *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [TEST.PD.EPR.SRC3.9#9]
14. Tester waits for 100ms for UUT to initiate the *EPR\_Mode* Message with Action field set 0x5(Exit) and Data field set to 0, else the Tester sends the *EPR\_Mode* Message with Action field set 0x5(Exit) and Data field set to 0
  15. Tester checks that the UUT sends *Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* (Exit) Message [TEST.PD.EPR.SRC3.9#10]
  16. Tester checks that the UUT completes the SPR contract negotiation using common procedure COMMON.PROC.PD.11 [TEST.PD.EPR.SRC3.9#13]

#### TEST.PD.EPR.SRC3.10 SPR mode - EPR\_Get\_Source\_Cap message

Description: As a Sink, the Tester verifies that the Source UUT responds with *Hard Reset* to the *EPR\_Request* Message received in the SPR\_Mode.

Test Specific Tester Behavior:

Test Conditions:

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (set EPR mode bit in RDO to 1).
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester behaves as a Provider/Consumer, and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
2. The Tester sends an *EPR\_Get\_Source\_Cap* Message.
3. The Tester checks the response. [TEST.PD.EPR.SRC3.10#1]
  - a. The check fails if the VIF field *EPR\_Supported\_As\_Src* is set to Yes and the UUT does not respond with a *EPR\_Source\_Capabilities* Message.
  - b. If the UUT is Provided Only and VIF field *EPR\_Supported\_as\_SRC* is No, the Tester checks that the UUT responds with the *Not\_Supported* Message and the test stops here.
  - c. If the UUT is DRP or C/P and VIF field *EPR\_Supported\_as\_SNK* is No and VIF field *EPR\_Supported\_as\_SRC* = No, the Tester checks that the UUT responds with the *Not\_Supported* Message and the test stops here.
  - d. If the UUT is not Provider Only and VIF field *EPR\_Supported\_as\_SRC* is No and VIF field *EPR\_Supported\_as\_Snk* is Yes, the Tester checks that the UUT responds with the *Reject* or *Not\_Supported* Message and the test stops here.
4. The Tester sends an *EPR\_Request* with Object Position set to 1.
5. The Tester checks the response. [TEST.PD.EPR.SRC3.10#2]
  - a. The check fails if the UUT does not send a *Hard\_Reset* Signaling within *tProtErrHardReset*, this timing is measured from the last bit of the *GoodCRC* of *EPR\_Request* to the last bit of the *Hard\_Reset* signal.

#### TEST.PD.EPR.SRC3.11 EPR Mode Exit by EPR\_Mode\_Exit message

Description: As a Sink, the Tester verifies that the Source UUT responds with *Hard\_Reset* to the *EPR\_Mode* (Exit).

Test Specific Tester Behavior:

Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Src* is set to No

-	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓

### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 (set EPR mode bit in RDO to 1).
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
2. The Tester sends an *EPR\_Mode* ( Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W).
3. The Tester checks that UUT responds *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0, otherwise the check fails and stops here [TEST.PD.EPR.SRC3.11#2]
4. If VIF field Captive\_Cable = Yes, skip to step 8, else Tester checks that UUT does not sends a *VCONN\_Swap* message [TEST.PD.EPR.SRC3.11#3]
5. If a SOP' *Discover Identity* Command is received from the UUT, the Tester replies as Passive Cable with V<sub>BUS</sub> Current Handling Capability set to 10b(5A), Maximum V<sub>BUS</sub> Voltage set to 11b(50V) and EPR Mode Capable field set to 1b (EPR Mode Capable)
6. The Tester checks if the UUT has sent SOP' *Discover Identity* Command any time prior sending the *EPR\_Mode*(Enter\_Succeeded) Message[TEST.PD.EPR.SRC3.11#4]
7. Tester checks that UUT completes the EPR entry process by sending *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) and Data field set to 0x0 within *tEnterEPR*, this timing is measured from the last bit of the *GoodCRC* in response to the *EPR\_Mode* (Enter) Message to the last bit of the *EPR\_Mode* ( Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.11#5]
8. Tester checks that the UUT sends *EPR\_Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* ( Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.11#6]
9. Tester checks that the UUT establishes EPR contract using common procedure COMMON.PROC.PD3.2 [TEST.PD.EPR.SRC3.11#8]



10. Tester sends a *EPR\_Mode* Message with the Action field set to 0x05(Exit) and Data field set to 0x00
11. The check fails if the UUT does not send a *Hard Reset* Signaling within *tProtErrHardReset*, this timing is measured from the last bit of the *GoodCRC* of *EPR\_Mode* ( Exit) Message to the last bit of the Hard\_Reset signal  
[TEST.PD.EPR.SRC3.11#7]

TEST.PD.EPR.SRC3.12 EPR mode - Get\_Source\_Cap message and *Request* Message response.

Description: As a Sink, the Tester verifies that the Source UUT responds with *Hard Reset* to the *Request* Message in the EPR mode.

Test Specific Tester Behavior:

Test Conditions:

This test is not applicable if the VIF parameter EPR\_Supported\_As\_Src is set to No

-	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓
Rev3UnchkdSrc		✓	✓

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
2. The Tester sends an *EPR\_Mode* ( Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)
3. The Tester checks that UUT responds *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0, otherwise the check fails and stops here [TEST.PD.EPR.SRC3.12#2]
4. If VIF field Captive\_Cable = Yes, skip to step 8, else Tester checks that UUT does not send a *VCONN\_Swap* message [TEST.PD.EPR.SRC3.12#3]

5. If a SOP' *Discover Identity* Command is received from the UUT, the Tester replies as Passive Cable with  $V_{BUS}$  Current Handling Capability set to 10b(5A), Maximum  $V_{BUS}$  Voltage set to 11b(50V) and EPR Mode Capable field set to 1b (EPR Mode Capable)
6. The Tester checks if the UUT has sent SOP' *Discover Identity* Command any time prior sending the *EPR\_Mode*(Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.12#4]
7. Tester checks that UUT completes the EPR entry process by sending *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) and Data field set to 0x0 within *tEnterEPR*, this timing is measured from the last bit of the *GoodCRC* in response to the *EPR\_Mode* (Enter) Message to the last bit of the *EPR\_Mode* (Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.12#5]
8. Tester checks that the UUT sends *EPR\_Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* ( Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.12#6]
9. Tester checks that the UUT establishes EPR contract using common procedure COMMON.PROC.PD3.2[TEST.PD.EPR.SRC3.12#9]
10. Tester sends a *Get\_Source\_Cap* Message.
11. The check fails if UUT doesn't send a *Source\_Capabilities* Message [TEST.PD.EPR.SRC3.12#7]
12. Tester responds with a *Request* Message with Object Position set to 1 and other fields set to values mentioned in bring-up procedure BU1.
13. Tester checks that the UUT sends a *Hard Reset* Signaling within *tProtErrHardReset*(this timing is measured from the last bit of the *GoodCRC* of *Request* Message to the last bit of the *Hard Reset* Signal)[TEST.PD.EPR.SRC3.12#8]

#### TEST.PD.EPR.SRC3.13 EPR mode - tSourceEPRKeepAlive Timeout

Description: The Tester as a sink does not send EPRKeepAlive message in EPR mode and verifies that the UUT sends *Hard Reset* Signaling within *tSourceEPRKeepAlive* timer

Test Specific Tester Behavior:

Test Conditions:

This test is not applicable if the VIF parameter EPR\_Supported\_As\_Src is set to No

-	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓

Rev3UnchkdSrc		✓	✓
---------------	--	---	---

### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester behaves as a Provider/Consumer and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
2. The Tester sends an *EPR\_Mode* ( Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)
3. The Tester checks that UUT responds *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0, otherwise the check fails and stops here [TEST.PD.EPR.SRC3.13#2]
4. If VIF field Captive\_Cable = Yes, skip to step 8, Tester checks that UUT does not sends a *VCONN\_Swap* Message [TEST.PD.EPR.SRC3.13#3]
5. If a SOP' *Discover Identity* Command is received from the UUT, the Tester replies as Passive Cable with V<sub>BUS</sub> Current Handling Capability set to 10b(5A), Maximum V<sub>BUS</sub> Voltage set to 11b(50V) and EPR Mode Capable field set to 1b (EPR Mode Capable)
6. The Tester checks if the UUT has sent SOP' *Discover Identity* Command any time prior sending the *EPR\_Mode*(Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.13#4]
7. Tester checks that UUT completes the EPR entry process by sending *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) and Data field set to 0x0 within *tEnterEPR*, this timing is measured from the last bit of the *GoodCRC* in response to the *EPR\_Mode* (Enter) Message to the last bit of the *EPR\_Mode* ( Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.13#5]
8. Tester checks that the UUT sends *EPR\_Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* ( Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.13#6]
9. Tester checks that the UUT establishes EPR contract using common procedure COMMON.PROC.PD3.2[TEST.PD.EPR.SRC3.13#8]
10. Tester doesn't send EPRKeepAlive message and checks that the UUT sends *Hard Reset* Signaling between *tSourceEPRKeepAlive* min(0.75sec) and max(1sec)), this timing is measured from the last bit of the *GoodCRC* in response to the last message

in the last AMS sequence to the first bit of the *Hard Reset* Signaling  
[TEST.PD.EPR.SRC3.13#7]

#### TEST.PD.EPR.SRC3.14 EPR mode - EPR\_Request with incorrect copy of PDO

Description: The Tester as a sink verifies the UUT responds with *Hard Reset* Signaling for *EPR\_Request* with incorrect copy of PDO.

Test Specific Tester Behavior:

Test Conditions:

This test is applicable if the VIF parameter EPR\_Supported\_As\_Src is set to Yes.

-	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSrc		✓	✓
Rev3UnchkdSrc		✓	✓

Test Procedures:

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
- The check fails if the "EPR mode capable" field is set to 0 in the *Source\_Capabilities* Message sent by UUT during the bring up procedure and the test stops here.  
[TEST.PD.EPR.SRC3.14#1]
- The Tester sends an *EPR\_Mode*(Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W) and enters EPR mode using common procedure COMMON.PROC.PD3.5. [TEST.PD.EPR.SRC3.14#2]
- Tester sends an *EPR\_Request* Message with Object Position set to 1, Operating Current and Maximum Operating Current set to 100mA and copy of PDO field set to the second PDO of the latest *EPR\_Source\_Capabilities* Message.
- The check fails if the UUT does not respond with the *Hard Reset* Signaling within *tProtErrHardReset*, which is measured from the last bit of the *GoodCRC* of the *EPR\_Request* Message to the first bit of the *Hard Rest* Signal  
[TEST.PD.EPR.SRC3.14#3]

#### TEST.PD.EPR.SRC3.15 DiscoverIdentityCounter and DiscoverIdentityTimer check for SOP'.

Description: The tester checks that a Source maintains *DiscoverIdentityCounter* and *DiscoverIdentityTimer* during an explicit contract. The tester intentionally does not reply with a *GoodCRC* for *SOP' Discover Identity* to ensure Source sends the *SOP' Discover Identity* Command requests every *tDiscoverIdentity* for no more than *nDiscoverIdentityCount*.

#### Test Specific Tester Behavior:

1. The tester does not respond to *SOP' Discover Identity* with *GoodCRC*.
2. The tester shall save the details of last received *nRetryCount* number of *SOP' Discover Identity* Commands.
3. Once an Explicit contract is established the tester maintains the overall count of *SOP' Discover Identity* received (*nDiscoverIdentity* and *nRetryCount* including) till the end of the test.

#### Test Conditions:

-	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3UnchkdSrc		✓	✓

#### Check Applicability:

The test is applicable if the VIF Parameter *EPR\_Supported\_As\_Src* is set to Yes and *Captive\_Cable* is set to No.

#### Test Procedures:

1. During the below bring-up procedures, if the UUT sends *SOP' Discover Identity* Command before establishing Explicit contract, record this fact for use in step 3.
2. There are 2 possible bring-up procedures:
  - a) The UUT has VIF field *PD\_Port\_Type* set to Provider Only, Provider/Consumer or DRP, the Tester behaves as a Sink Only and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
    1. The check fails if the "EPR mode capable" field is set to 0 in the *Source\_Capabilities* Message sent by UUT during the bring up procedure and the test stops here. [TEST.PD.EPR.SRC3.15#1]
    2. The Tester sends a *Request* Message with EPR Mode Capable Bit, B22, set to 1.
  - b) The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider, the Tester behaves as a Provider/Consumer, and it runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7
    1. The check fails if the "EPR mode capable" field is set to 0 in the *Source\_Capabilities* Message sent by UUT during the bring up procedure and the test stops here. [TEST.PD.EPR.SRC3.15#1]
    2. The Tester sends a *Request* Message with EPR Mode Capable Bit, B22, set to 1.
3. The Tester waits to see if it receives an *SOP' Discover Identity* Command within *tEPRSourceCableDiscovery* of receiving the *GoodCRC* Message acknowledging the *PS\_RDY* Message as part of the Explicit Contract negotiation.
  - b. If it does not, and no *SOP' Discover Identity* was seen before the explicit contract, the check fails [TEST.PD.EPR.SRC3.15#2]
  - c. If it does not, the test stops here.
4. If the received *SOP' Discover Identity* Request is the first such request after the explicit contract, wait for the next *SOP' Discover Identity* Request.

Else, do the following checks:

- a) The Tester checks the total number of the *SOP' Discover Identity* Commands, if it is more than  $nDiscoverIdentityCount \times (nRetryCount + 1)$ , the check fails, and the test stops here. [TEST.PD.EPR.SRC3.15#3]
- b) If the message ID of the current *SOP' Discover Identity* Request received is the same Message ID as previous the *SOP' Discover Identity* Command:
  1. The Tester measures the delay between the current and the previous *SOP' Discover Identity*.
  2. The test fails if the delay is not between  $tReceive.min$  and  $tReceive.max$  [TEST.PD.EPR.SRC3.15#4]
- c) If the message ID of the current *SOP' Discover Identity* Command is not the same as the previous *SOP' Discover Identity* Command:
  1. The tester measures the delay between the current *SOP' Discover Identity* Command and the first *SOP' Discover Identity* Command received with the previous Message ID.
  2. The test fails if the delay is not between  $tDiscoverIdentity.min$  and  $tDiscoverIdentity.max$  [TEST.PD.EPR.SRC3.15#5]
5. Repeat step 4, if the UUT is sending the *SOP' Discover Identity* Command, else the test stops.

#### TEST.PD.EPR.SRC3.16 PR\_Swap for the UUT as EPR Source

Description: The Tester as a Source checks that the UUT rejects the PR\_SWAP Message in the EPR mode.

#### Test Conditions:

-	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3UnchkdSrc		✓	✓

#### Check Applicability:

The test is applicable if the VIF Parameter EPR\_Supported\_As\_Src is set to Yes

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The check fails if the "EPR mode capable" field is set to 0 in the *Source Capabilities* Message sent by the UUT during the bring up procedure and the test stops here. [TEST.PD.EPR.SRC3.16#1]
3. The Tester sends an *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter) and Data field set to the power in watts specified by the VIF PD\_Power\_As\_Source/1000, and enters the EPR mode using common procedure COMMON.PROC.PD3.5. [TEST.PD.EPR.SRC3.16#2]

4. The Tester checks that the UUT sends the *EPR\_Source\_Capabilities* Message within *tFirstSourceCap* of the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* (Enter\_Succeeded) Message [TEST.PD.EPR.SRC3.16#3]
5. The Tester checks that the UUT establishes EPR contract using common procedure COMMON.PROC.PD3.2 [TEST.PD.EPR.SRC3.16#4]
6. The Tester sends the PR\_Swap Message.
7. The Tester checks that the UUT responds with [TEST.PD.EPR.SRC3.16#5]
  - a. *Not\_Supported* if the UUT is Provider Only
  - b. Otherwise, *Reject*

#### 7.1.2 PD3 Mode – EPR\_Power\_Supply Specific Tests

The tests in this subsection are run in PD3 mode.

The tests in this subsection are applicable to UUT with VIF field PD\_Port\_Type set to Provider Only, Provider/Consumer, Consumer/Provider and DRP.

Unless a Test Specific Tester Behavior is specified, this section uses the following behavior:

- The Tester is a Sink
- The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always rejects *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### TEST.PD.PS.EPR.SRC3.1 Multiple EPR Request Load Test

Description: As a Sink only, the Tester verifies that the Source UUT responds correctly to various *EPR\_Request* Messages.

Test Specific Tester Behavior:

Test Conditions:

This test is applicable if the VIF parameter EPR\_Supported\_As\_Src is set to Yes.

	Consumer Only	Provider Only, P/C, C/P, DRP
<i>Rev3ChkdSrc</i>		✓

Test Requirements: If Captive\_Cable = YES, *vSrcNew* min or *vAvsNew* min limit shall be (*vSrcNew* min) - (0.75\*I/X) or (*vAvsNew* min) - (0.75\*I/X), where I is the actual current being drawn, and X=3 if the sourced PDO contract is within the range (0, 3A], or X=5 if the sourced PDO contract is within the range (3A, 5A)



### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The check fails if the "EPR mode capable" field is set to 0 in the *Source\_Capabilities* Message sent by UUT during the bring up procedure and the test stops here. [TEST.PD.PS.EPR.SRC3.1#1]
3. The Tester sends an *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W) and enters EPR mode using common procedure COMMON.PROC.PD3.5. [TEST.PD.PS.EPR.SRC3.1#2]
4. The Tester repeats the following steps for each of the EPR\_PDOs only (in the *EPR\_Source\_Capabilities* Message), starting with the 8th PDO:
  - a. The Tester sends an *EPR\_Request* Message.
    - i. B27 (Reserved) = 0b
    - ii. B26 (Capability Mismatch) = 0b
    - iii. B25 (USB Communication Capable) = 0b
    - iv. B24 (No USB Suspend) = 1b
    - v. B22 (EPR Mode Capable) = 1b
    - vi. For Fixed PDO:
      1. B19...10 (Operating Current):
        - a. For first *EPR\_Request*: 0mA
        - b. For subsequent Request: As defined in Step 4.f.i
      2. B9...0 (Maximum Operating Current) = Maximum Current in the Source PDO.
    - vii. For EPR AVS APDO:
      1. B20...9 (Output Voltage) = Maximum Voltage offered in Source APDO
      2. B6...0 (Operating Current):
        - a. For the first *EPR\_Request*: 0mA
        - b. For subsequent Request: As defined in Step 4.e.i

The initial *EPR\_Request* is for PDO #8 (28V Fixed) at 0mA. The complete list of *EPR\_Request* is defined in step 4.e.

  - b. The check fails if either *Accept*, *Reject* or *Wait* Message is not received, and the test stops here. If the *Reject* Message is received, the Tester reports the check with a warning, and the test stops here. [TEST.PD.PS.EPR.SRC3.1#11]
  - c. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after *nCtsWaitCount* retries, and the test stops here. [TEST.PD.PS.EPR.SRC3.1#13]
  - d. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [TEST.PD.PS.EPR.SRC3.1#12]

- e. The Tester performs the following checks:
- i. If the transition involves a current decrease (but no PDO change), the Tester decreases current to the new value within *tSnkNewPower* min of the last bit of the *GoodCRC* sent in response to the *Accept* Message. The Tester draws the requested Operating Current with a current transition slew rate of 100mA per  $\mu$ s. The Tester monitors  $V_{BUS}$  voltage and the check fails if  $V_{BUS}$  voltage is not within the limits of *vSrcNew* or *vAvsNew*. [TEST.PD.PS.EPR.SRC3.1#3]
  - ii. If the transition involves a current increase (but no PDO change), the Tester increases current to the new value after the last bit of the *GoodCRC* sent in response to the *PS\_RDY* Message. The Tester draws the requested Operating Current with a current transition slew rate of 100mA per  $\mu$ s. The Tester monitors  $V_{BUS}$  voltage and the check fails if  $V_{BUS}$  voltage is not within the limits of *vSrcNew* or *vAvsNew*. [TEST.PD.PS.EPR.SRC3.1#4]
  - iii. If the transition involves a change of PDO with a voltage increase, decrease the current drawn by the Tester to less than *iSnkStdby* within *tSnkStdby* of the last bit of the *GoodCRC* sent in response to the *Accept* Message. If the transition involves a change of PDO with decreasing voltage and current, the Tester decreases its current draw to the requested current or power for Fixed, Battery, Variable and SPR\_AVs PDOs, 80% of the requested current for PPS PDO within *tSrcTransition* min after the end of the *GoodCRC* sent in response to the *Accept* Message. The Tester performs the following checks:
    1. The Tester checks that  $V_{BUS}$  remains within *vSrcNew* or *vAvsNew* of the starting voltage within *tSrcTransition* min after the end of the *GoodCRC* sent in response to the *Accept* Message. [TEST.PD.PS.EPR.SRC3.1#5]
    2. The Tester checks that the  $V_{BUS}$  transition meets *vSrcSlewPos* and *vSrcSlewNeg*. [TEST.PD.PS.EPR.SRC3.1#6]. (See **Appendix E. Slew Rate Measurements.** )
    3. During the voltage transition, the Tester checks that the  $V_{BUS}$  voltage remains within *vSrcValid* limits from the time of crossing into the *vSrcValid* limits until *tSrcSettle* max. [TEST.PD.PS.EPR.SRC3.1#7]
    4. The Tester checks that the  $V_{BUS}$  voltage remains within *vSrcNew* or *vAvsNew* limits between *tSrcSettle* max and 100ms after *tSrcSettle* max. [TEST.PD.PS.EPR.SRC3.1#8]
    5. The Tester checks that the UUT does not send *PS\_RDY* before  $V_{BUS}$  is within *vSrcNew* or *vAvsNew*. [TEST.PD.PS.EPR.SRC3.1#9]
    6. After having sent *GoodCRC* in response to *PS\_RDY*, if the transition involves a current increase, the Tester increases its current draw to the current requested and The Tester checks that

the  $V_{BUS}$  voltage remains within *vSrcNew* or *vAvsNew* limits for 100ms after receiving *PS\_RDY* [TEST.PD.PS.EPR.SRC3.1#10]

- f. The Tester repeats Step-4a to Step-4d, replacing the appropriate fields with the next value from the following list. The Tester sends 9 separate *EPR\_Request* Messages for each *EPR\_PDO*.
  - i. For Fixed PDOs, replace B19...10 (Operating Current) and For EPR AVS APDO, replace B6...0 (Operating Current):
    1. 0mA [performed in Step 4.b; listed here for reference only]
    2. 25% of Maximum Current
    3. 50% of Maximum Current
    4. 75% of Maximum Current
    5. 100% of Maximum Current
    6. 75% of Maximum Current
    7. 50% of Maximum Current
    8. 25% of Maximum Current
    9. 0mA

#### TEST.PD.PS.EPR.SRC3.2 PDO Transitions in EPR Mode

Description: As a Sink only, the Tester enumerates all possible transitions among the PDOs of the UUT and verifies the UUT performs PDO transitions correctly in EPR Mode.

Test Specific Tester Behavior: N/A

Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Src* is set to No

	Consumer Only	Provider Only, P/C, C/P, DRP
<i>Rev3ChkdSrc</i>		✓

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field *PD\_Port\_Type* set to anything else, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The check fails if the "EPR mode capable" field is set to 0 in the *Source\_Capabilities* Message sent by UUT during the bring up procedure and the test stops here. [TEST.PD.PS.EPR.SRC3.2#1]
3. The Tester sends an *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W) and enters EPR mode using common procedure COMMON.PROC.PD3.5[TEST.PD.PS.EPR.SRC3.2#2]
4. The Tester repeats the following steps using the PDO as provided in the *EPR\_PDO* Sequence Table (see Table 15). **NOTE:** PDO Transitions only applicable for valid PDOs

in the *EPR\_Source\_Capabilities*. Ignore the unused PDOs. For AVS APDO, request the max, min, and mid voltages in sequence.

- a. The Tester sends an *EPR\_Request* Message:
  - i. B27 (Reserved) = 0b
  - ii. B26 (Capability Mismatch) = 0b
  - iii. B25 (USB Communication Capable) = 0b
  - iv. B24 (No USB Suspend) = 1b
  - v. B22 (EPR Mode Capable) = 1b
  - vi. For Fixed PDO:  
B19...10 (Operating Current) = 100mA  
B9...0 (Maximum Operating Current) = 100mA
  - For EPR AVS PDO:  
B20...9 (Output Voltage) = Maximum Voltage offered in Source APDO  
B7...0 (Operating Current) = 100mA
- b. The check fails if either *Accept*, *Reject* or *Wait* Message is not received, and the test stops here. If *Reject* Message is received, the Tester reports the check with a warning, and the test stops here. [TEST.PD.PS.EPR.SRC3.2#3]
- c. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for *nCtsWaitCount* times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after *nCtsWaitCount* retries. [TEST.PD.PS.EPR.SRC3.2#10]
- d. The check fails if the first bit of Preamble of the *PS\_RDY* Message is not received within *tSrcTransReq* from the last bit of the *GoodCRC* Message EOP corresponding to a transmitted *Accept* Message. [TEST.PD.PS.EPR.SRC3.2#4]
- e. The Tester performs the following checks:  
Decrease the current drawn by the Tester to less than *iSnkStdby* within *tSnkStdby* min of the last bit of the *GoodCRC* sent in response to the *Accept* Message. The Tester performs the following checks:
  1. The Tester checks the  $V_{BUS}$  is within *vSrcNew* or *vAvsNew* of the starting voltage within *tSrcTransition* min after the end of the *GoodCRC* sent in response to the *Accept* Message. [TEST.PD.PS.EPR.SRC3.2#5]
  2. The Tester checks that the  $V_{BUS}$  transition meets *vSrcSlewPos* and *vSrcSlewNeg*. [TEST.PD.PS.EPR.SRC3.2#6] (See **Appendix E. Slew Rate Measurements.** )
  3. During the voltage transition, the Tester checks that the  $V_{BUS}$  voltage remains within *vSrcValid* limits from the time of crossing into the *vSrcValid* limits until *tSrcSettle* max. [TEST.PD.PS.EPR.SRC3.2#7]
  4. The Tester checks that the  $V_{BUS}$  voltage is within *vSrcNew* or *vAvsNew* limits between *tSrcSettle* max and 100ms after *tSrcSettle* max. [TEST.PD.PS.EPR.SRC3.2#8]

5. The Tester checks that the UUT does not send *PS\_RDY* before the  $V_{BUS}$  is within *vSrcNew* or *vAvsNew*.  
[TEST.PD.PS.EPR.SRC3.2#9]

## 7.2 Test Procedure for EPR Sink Capable UUT

### 7.2.1 PD3 Mode

The tests in this subsection are run in PD3 mode.

Cable Discovery message sequences are not explicitly mentioned in all the tests in this subsection, as it can be varied based on the test tool implementation and test cable used in the test setup.

Ensure the testing environment favors the UUT to initiate the *EPR\_Mode* (Enter) Message after the SPR contract negotiation to successfully validate the EPR negotiation

- This section uses the following behavior, except where defined in Test Specific Tester Behavior or a Test Procedure: The Tester supports both Power Roles but does not toggle (ie, a Provider/Consumer or Consumer/Provider) The Tester always replies with an *Accept* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always rejects *VCONN\_Swap* COMMON.PROC.PD.6, unless mentioned otherwise in the test procedure
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8
- The Tester sets the *EPR\_Mode\_Capable* bit to 1 in the *Source\_Caps* and *EPR\_Source\_Capabilities* Message, unless mentioned otherwise in the test procedure
- The Tester always sets 4 PDOs (5V @ 3A, 9V @ 3A, 15V @ #A and 20V @ 5A) in the *Source\_Capabilities* Message unless otherwise specified in the test procedure.  
(Applicable for COMMON.PROC.BU.2 and COMMON.PROC.BU.8)

### TEST.PD.EPR.SNK3.1 EPR Entry Process - Success

Description: The Tester as a source verifies the successful EPR negotiation

Test Specific Tester Behavior:

Test Conditions:

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSnk	✓		✓
Rev3UnchkdSnk	✓		✓

## Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer /Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends *EPR\_Get\_Sink\_Cap* Message and perform below check [TEST.PD.EPR.SNK3.1#11]
  - a. If the UUT is Consumer Only and VIF field EPR\_Supported\_as\_SNK is No, the Tester checks that the UUT responds with the *Not\_Supported* Message.
  - b. If the UUT is DRP or C/P or P/C and VIF field EPR\_Supported\_as\_SNK is No and VIF field EPR\_Supported\_as\_SRC is No , the Tester checks that the UUT responds with the *Not\_Supported* Message
  - c. If the UUT is not Consumer Only and VIF field EPR\_Supported\_as\_SNK is No and VIF field EPR\_Supported\_as\_SRC is Yes, the Tester checks that the UUT responds with the *Reject* or *Not\_Supported* Message
  - d. If the UUT VIF field EPR\_Supported\_as\_SNK is Yes, the Tester checks that the UUT responds with the *EPR\_Sink\_Capabilities* Message.

*NOTE:* *EPR\_Sink\_Capabilities* should not be the maximum power the UUT is able to request
3. Tester waits for 5 seconds and performs below checks, [TEST.PD.EPR.SNK3.1#2]
  - a. If the VIF field EPR\_Supported\_As\_Snk is No, then the check fails if the UUT sends an *EPR\_Mode* Message. The test stops here.
  - b. If the VIF field EPR\_Supported\_As\_Snk is Yes and the UUT does not send an *EPR\_Mode* Enter message with the Action field set to 0x01, the test stops here, and the result is a warning.
4. If VIF field Captive\_Cable = Yes, the Tester sends SOP' *Discover Identity* Command and checks the below fields in the response message, [TEST.PD.EPR.SNK3.1#3]
  - e. V<sub>BUS</sub> Current Handling Capability set to 10b(5A)
  - f. Maximum V<sub>BUS</sub> Voltage set to 11b(50V)
  - g. EPR Mode Capable field set to 1b (EPR Mode Capable)



5. The Tester sends the *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) and the Data field set to 0x0 within *tEnterEPR* from the *EPR\_Mode* ( Enter) Message.
6. The Tester sends the *EPR\_Source\_Capabilities* Message after 50ms from the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* ( Enter\_Succeeded) Message and establishes EPR contract using common procedure COMMON.PROC.PD3.3 from Step2. [TEST.PD.EPR.SNK3.1#10]
7. The Tester waits for 10 seconds and if there are no messages initiated by the UUT for more than *tSinkEPRKeepAlive* max, then the check fails [TEST.PD.EPR.SNK3.1#4]
8. If the UUT does not send even a single the *EPR\_KeepAlive* Message in the above step, make this check as warning [TEST.PD.EPR.SNK3.1#5]
9. The Tester re-sends the *EPR\_Source\_Capabilities* Message as per COMMON.PROC.PD3.3 Step2.b. The check fails if the UUT does not respond with an *EPR\_Request* Message [TEST.PD.EPR.SNK3.1#13]
10. The Tester intentionally sends a *Wait* Message. The Tester shall continue monitoring for *tSourceEPRKeepAlive* max that the UUT is not disconnecting. Disconnect is deemed to have taken place when the CC line voltage falls outside the ‘connect’ range, as a result of Rd being released by the UUT. If the UUT sends another *EPR\_Request* Message during this time, the Tester will respond with a *Wait* Message. [TEST.PD.EPR.SNK3.1#14]
11. If there are no messages initiated by the UUT for more than *tSinkEPRKeepAlive* max, then the check fails [TEST.PD.EPR.SNK3.1#12]
12. The Tester waits for 1 second and if the UUT does not send the *EPR\_Request* Message with SPR RDO, the Tester sends the *EPR\_SourceCaps* Message with no EPR PDO's and checks that the UUT responds with the *EPR\_Request* Message [TEST.PD.EPR.SNK3.1#6]
13. The Tester checks that the UUT establishes the EPR contract negotiation using COMMON.PROC.PD3.4 [TEST.PD.EPR.SNK3.1#8]
14. The Tester waits for 100ms for UUT to initiate the *EPR\_Mode* Message with Action field set 0x5(Exit) and Data field set to 0, else the Tester sends the *EPR\_Mode* Message with Action field set 0x5(Exit) and Data field set to 0
15. The Tester sends the *Source\_Capabilities* Message and checks that the UUT responds with *Request* Message [TEST.PD.EPR.SNK3.1#7]
16. The Tester checks that the UUT establishes the SPR contract negotiation using COMMON.PROC.PD.10 [TEST.PD.EPR.SNK3.1#9]



## TEST.PD.EPR.SNK3.2 EPR Entry Failed - tEnterEPR timeout

Description: The Tester as a source does not send *EPR\_Mode* Enter Succeeded message in the EPR mode sequence and verifies that the UUT sends *Soft Reset* Message.

Test Specific Tester Behavior:

Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Snk* is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSnk	✓		✓

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer /Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The check fails if the "EPR mode capable" field is set to 0 in the *Request* Message sent by UUT during the bring up procedure and the test stops here [TEST.PD.EPR.SNK3.2#1]
3. Tester waits for 5 seconds for the UUT to initiate the *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter) , otherwise the test passes and stops here.[TEST.PD.EPR.SNK3.2#2]
4. Tester responds with the *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.
5. If VIF field Captive\_Cable = Yes, the Tester sends SOP' *Discover Identity* Command and checks the below fields in the response message [TEST.PD.EPR.SNK3.2#3]
  - a. V<sub>BUS</sub> Current Handling Capability set to 10b(5A)
  - b. Maximum V<sub>BUS</sub> Voltage set to 11b(50V)
  - c. EPR Mode Capable field set to 1b (EPR Mode Capable)
6. Tester doesn't send the *EPR\_Mode* (Enter\_Succeeded) Message.

7. Tester checks that UUT starts to send Soft Reset between *tEnterEPR* min and max from the EOP of the *GoodCRC* Message of the *EPR\_Mode* ( Enter) Message.[TEST.PD.EPR.SNK3.2#4]

#### TEST.PD.EPR.SNK3.3 EPR Entry Failed by the *EPR\_Mode* ( Enter failed) Message

Description: :As a Source, the Tester verifies that the UUT responds with a *Soft Reset* Message to the *EPR\_Mode* Failed message.

#### Test Specific Tester Behavior:

#### Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Snk* is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSnk	✓		✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer /Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The check fails if the "EPR mode capable" field is set to 0 in the *Request* Message sent by UUT during the bring up procedure and the test stops here [TEST.PD.EPR.SNK3.3#1]
3. Tester waits for 5 seconds for the UUT to initiate the *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter), otherwise the test passes and stops here [TEST.PD.EPR.SNK3.3#2]
4. Tester responds with the *EPR\_Mode* Message with Action field set to 0x04(Enter Failed) and Data field set to 0x03(EPR Mode Capable bit not set in RDO).
5. Tester checks that UUT sends a *Soft Reset* Message within *tProtErrSoftReset*, this timing is measured from the last bit of the *GoodCRC* Message of the *EPR\_Mode*

(Enter Failed) Message to the last bit of the *Soft Reset* Message).  
[TEST.PD.EPR.SNK3.3#3]

6. If UUT initiates the *EPR\_Mode* (Enter) Message in step3, then repeat this test for all conditions in the Data field and include 1 reserved value (6), otherwise the test stops here.

#### TEST.PD.EPR.SNK3.4 EPR Fail due to tFirstSourceCap timeout.

Description: The Tester as a source verifies the UUT responds with *Hard Reset* for Source Capability timeout by Tester

Test Specific Tester Behavior:

Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Snk* is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSnk	✓		✓
Rev3UnchkdSnk	✓		✓

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer /Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The check fails if the "EPR mode capable" field is set to 0 in the *Request* Message sent by UUT during the bring up procedure and the test stops here  
[TEST.PD.EPR.SNK3.4#1]
3. Tester waits for 5 seconds for the UUT to initiate the *EPR\_Mode* ( Enter) Message with the Action field set to 0x01(Enter) , otherwise the test passes and stops here  
[TEST.PD.EPR.SNK3.4#2]
4. Tester responds with the *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.

5. Tester sends the *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) after 100ms from the *EPR\_Mode* (Enter) Message.
6. Tester doesn't send an *EPR\_Source\_Capabilities* Message
7. The check fails if the UUT doesn't send a *Hard Reset* Signaling within *tTypeCSinkWaitCap* (this timing is measured from the last bit of the *GoodCRC* Message of *EPR\_Mode* (Enter\_succeeded) to the last bit of the *Hard Reset* Signal)[TEST.PD.EPR.SNK3.4#3]

#### TEST.PD.EPR.SNK3.5 EPR Exit due to incorrect EPR Source Cap

**Description:** The Tester as a source verifies the EPR Exit due to the incorrect EPR Source Capability

**Test Specific Tester Behavior:**

**Test Conditions:**

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Snk* is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSnk	✓		✓

**Test Procedures:**

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer /Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The check fails if the "EPR mode capable" field is set to 0 in the *Request* Message sent by UUT during the bring up procedure and the test stops here [TEST.PD.EPR.SNK3.5#1]
3. Tester waits for 5 seconds for the UUT to initiate the *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter) , otherwise the test passes and stops here [TEST.PD.EPR.SNK3.5#2]

4. Tester responds with the *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.
5. Tester sends the *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) after 100ms from the *EPR\_Mode*( Enter) Message
6. Tester sends the *EPR\_Source\_Capabilities* Message with PDO1 set to 5V 3A, PD02- 9V/3A, PDO3- 15V/3A, PDO4-20V/5A, PDO5 set to 28V 5A, PDO6 to PDO7 set to 0 and PDO8 set to 28V 5A.
7. The check fails if the UUT doesn't send a *Hard Reset* Signaling within *tProtErrHardReset*(this timing is measured from the last bit of the *GoodCRC* Message of the *EPR\_Source\_Capabilities* Message to the last bit of the *Hard Reset* Signal)[TEST.PD.EPR.SNK3.5#3]

#### TEST.PD.EPR.SNK3.6 EPR Exit due to *EPR\_Mode* Exit Message

Description: The Tester as a source verifies the UUT responds with *Hard Reset* for *EPR\_Mode* (Exit) Message (In the EPR PD Negotiation)

Test Specific Tester Behavior:

Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Snk* is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSnk	✓		✓

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer /Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The check fails if the "EPR mode capable" field is set to 0 in the *Request* Message sent by UUT during the bring up procedure and the test stops here [TEST.PD.EPR.SNK3.6#1]

3. Tester waits for 5 seconds for the UUT to initiate the *EPR\_Mode* ( Enter) Message with the Action field set to 0x01(Enter) , otherwise the test passes and stops here  
[TEST.PD.EPR.SNK3.6#2]
4. Tester responds with the *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.
5. Tester sends the *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) after 100ms from the *EPR\_Mode* ( Enter) Message.
6. Tester sends the *EPR\_Source\_Capabilities* Message after 50ms from the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* ( Enter\_Succeeded) Message and establishes EPR contract using common procedure COMMON.PROC.PD3.3 from Step2. [TEST.PD.EPR.SNK3.6#5]
7. Tester waits for 1 second and checks that UUT sends the *EPR\_KeepAlive* Message within *tSinkEPRKeepAlive* max from the EOP of the last message of the last AMS sequence [TEST.PD.EPR.SNK3.6#3]
8. If the Tester and UUT are still in an explicit contract in EPR mode with an EPR PDO then the Tester sends *EPR\_Mode* Message with Action field set to 5(Exit), else the test passes and stops here.
9. The check fails if the UUT doesn't sent a *Hard\_Reset* Signaling within *tProtErrHardReset*(this timing is measured from the last bit of the *GoodCRC* Message of the *EPR\_Mode* (Exit) Message to the last bit of the *Hard\_Reset* Signal)  
[TEST.PD.EPR.SNK3.6#4]

TEST.PD.EPR.SNK3.7 *EPR\_Fail\_by\_Wait\_Message* – Deprecated

#### TEST.PD.EPR.SNK3.8 *EPR Exit due to Source Cap*

**Description:** The Tester as a source verifies the UUT responds with *Hard\_Reset* for the Source Capability message after the EPR PD negotiation.

**Test Specific Tester Behavior:**

**Test Conditions:**

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Snk* is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSnk	✓		✓

## Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer /Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The check fails if the "EPR mode capable" field is set to 0 in the *Request* Message sent by UUT during the bring up procedure and the test stops here  
[TEST.PD.EPR.SNK3.8#1]
3. Tester waits for 5 seconds for the UUT to initiate the *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter), otherwise the test passes and stops here  
[TEST.PD.EPR.SNK3.8#2]
4. Tester responds with the *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.
5. Tester sends the *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) after 100ms from the *EPR\_Mode* (Enter) Message.
6. Tester sends the *EPR\_Source\_Capabilities* Message after 50ms from the last bit of the *GoodCRC* Message acknowledging the *EPR\_Mode* (Enter\_Succeeded) Message and establishes EPR contract using common procedure COMMON.PROC.PD3.3 from Step2.
7. Tester waits for 1 second and checks that UUT sends EPR\_KeepAlive message within *tSinkEPRKeepAlive* max from the EOP of the last message of the last AMS sequence  
[TEST.PD.EPR.SNK3.8#3]
8. Tester sends the Source Cap message.
9. Tester checks that the UUT sends a *Hard Reset* Signaling within *tProtErrHardReset*(this timing is measured from the last bit of the *GoodCRC* Message of the *Source\_Capabilities* Message to the last bit of the *Hard Reset* Signal)  
[TEST.PD.EPR.SNK3.8#4]

## TEST.PD.EPR.SNK3.9 EPR Entry failed due to SourceCap

Description: The Tester as a source verifies the UUT responds with *Hard Reset* for the *Source\_Capabilities* Message during the EPR PD negotiation.



### Test Specific Tester Behavior:

### Test Conditions:

This test is not applicable if the VIF parameter *EPR\_Supported\_As\_Snk* is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSnk	✓		✓

### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer /Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The check fails if the "EPR mode capable" field is set to 0 in the *Request* Message sent by UUT during the bring up procedure and the test stops here  
[TEST.PD.EPR.SNK3.9#1]
3. Tester waits for 5 second for the UUT to initiate the *EPR\_Mode* (Enter) Message with the Action field set to 0x01(Enter) , otherwise the test passes and stops here  
[TEST.PD.EPR.SNK3.9#2]
4. Tester responds with the *EPR\_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.
5. Tester sends the *EPR\_Mode* Message with Action field set to 0x03(Enter\_Succeeded) after 100ms from the *EPR\_Mode* ( Enter) Message.
6. Tester initiates the *Source\_Capabilities* Message after 50ms from the *EPR\_Mode* (Enter\_Succeeded) Message
7. Tester checks that the UUT sends a *Hard Reset* Signaling within *tProtErrHardReset*(this timing is measured from the last bit of the *GoodCRC* Message of the *Wait* Message to the last bit of the *Hard Reset* Signaling)  
[TEST.PD.EPR.SNK3.9#4]

*TEST.PD.EPR.SNK3.10 EPR Exit fail due to SinkWaitCapTimer timeout*

Description: The Tester as a source verifies the UUT responds with *Hard Reset* for *SinkWaitCapTimer* timeout by Tester during the EPR Exit process.

### Test Specific Tester Behavior:

### Test Conditions:

This test is not applicable if the VIF parameter `EPR_Supported_As_Snk` is set to No

	Consumer Only	Provider Only	DRP, C/P, P/C
<i>Rev3ChkdSnk</i>	✓		✓
<i>Rev3UnchkdSnk</i>	✓		✓

### Test Procedures:

1. There are 2 possible bring-up procedures:
  - c. The UUT has VIF field `PD_Port_Type` set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink `COMMON.PROC.BU.8`
  - d. The UUT has VIF field `PD_Port_Type` set to Consumer Only, Consumer /Provider or `DRP`, the Tester runs bring-up procedure with the UUT as a Sink `COMMON.PROC.BU.2`
2. The check fails if the "EPR mode capable" field is set to 0 in the *Request* Message sent by UUT during the bring up procedure and the test stops here.  
[TEST.PD.EPR.SNK3.10#1]
3. Tester waits for 5 seconds for the UUT to initiate the *EPR\_Mode* (Enter) Message with the Action field set to 1(Enter) and completes the EPR Entry process using `COMMON.PROC.PD3.6`, otherwise the test passes and stops here.  
[TEST.PD.EPR.SNK3.10#2]
4. If the Tester and UUT are still in an explicit contract in EPR mode with an EPR PDO then the Tester sends an *EPR\_Source\_Capabilities* Message with no EPR PDO's and checks that the UUT responds with the *EPR\_Request* Message, else go directly to step 6.  
[TEST.PD.EPR.SNK3.10#3]
5. Tester checks that the UUT establishes the EPR contract negotiation using `COMMON.PROC.PD3.4` [TEST.PD.EPR.SNK3.10#4]
6. If the Tester and UUT are in an explicit contract in EPR mode with SPR PDO then the Tester sends an *EPR\_Mode* Message with Action field set to 5(Exit) and Data field set to 0, else the test passes and stops here.  
Tester doesn't send the *Source\_Capabilities* Message and checks that the UUT responds with a *Hard\_Reset* Signaling within *tTypeCSinkWaitCap+tReceive*. This timing is measured from the last bit of the *GoodCRC* Message of *EPR\_Mode* (Exit) Message to the last bit of the *Hard\_Reset* Signal. [TEST.PD.EPR.SNK3.10#6]

*TEST.PD.EPR.SNK3.11 PR\_Swap for the UUT as the EPR Sink*

Description: The Tester as a Source checks that the UUT rejects the *PR\_Swap* Message in the EPR mode

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSnk	✓		✓

Check Applicability: The Test is applicable if the VIF Parameter `EPR_Supported_As_Snk` is set to Yes

Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field `PD_Port_Type` set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - b. The UUT has VIF field `PD_Port_Type` set to Consumer Only, Consumer /Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
2. The Tester sends *EPR\_Get\_Sink\_Cap* Message and checks that the UUT responds with the *EPR\_Sink\_Capabilities* Message. [TEST.PD.EPR.SNK3.11#2]
 

*NOTE:* *EPR\_Sink\_Capabilities* should not be the maximum power the UUT is able to request
3. The Tester waits for *tCtsInfinite*.
4. If VIF field `Captive_Cable` = Yes, the Tester sends SOP' *Discover Identity* Command and checks the below fields in the response message [TEST.PD.EPR.SNK3.11#4]
  - h.  $V_{BUS}$  Current Handling Capability set to 10b(5A)
  - i. Maximum  $V_{BUS}$  Voltage set to 11b(50V)
  - j. EPR Mode Capable field set to 1b (EPR Mode Capable)
5. The Tester sends the *EPR\_Mode* Message with the Action field set to 0x03(Enter\_Succeeded) and the Data field set to 0x0 within *tEnterEPR* from the *EPR\_Mode* (Enter) Message.
6. The Tester sends the *EPR\_Source\_Capabilities* Message that contains seven PDOs and four EPR four EPR\_PDOs (240W)
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 0b if Tester is a Source only; set to 1b if Tester is a Dual-Role Power
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B27 (Unconstrained Power) set to 1b.
  - e. B26 (USB Communications Capable) set to 0b.
  - f. B25 (Dual-Role Data) set to 0b.

- g. B24 (Unchunked Extended messages Supported) set to 0b or 1b depending on test condition.
  - h. 23 set to 1b (*EPR\_Mode* Capable)
  - i. 22 (Reserved) set to 0b
  - j. B21...20 (Peak Current) set to 00b.
    - PDO1: 9. B19...10 (Voltage) set to 5V.  
10. B9...0 (Maximum Current) set to 3000mA
    - PDO2: 9. B19...10 (Voltage) set to 9V.  
10. B9...0 (Maximum Current) set to 3000mA
    - PDO3: 9. B19...10 (Voltage) set to 15V.  
10. B9...0 (Maximum Current) set to 3000mA
    - PDO4: 9. B19...10 (Voltage) set to 20V.  
10. B9...0 (Maximum Current) set to 5000mA
    - PDO5: B31..30 (APDO) set to 11b  
B29..28 (SPR AVS) set to 10b  
B27..26 (Peak Current) set to 00b  
B25..20 (Reserved) set to 00b  
B19..10 (9-15V current) set to 3000mA  
B9..0 (15-20v current) set to 5000mA
    - PDO6: All fields set to 0.
    - PDO7: All fields set to 0.
    - PDO8: 9. B19...10 (Voltage) set to 28V.  
10. B9...0 (Maximum Current) set to 5000mA
    - PDO9: 9. B19...10 (Voltage) set to 36V.  
10. B9...0 (Maximum Current) set to 5000mA
    - PDO10: 9. B19...10 (Voltage) set to 48V.  
10. B9...0 (Maximum Current) set to 5000mA
    - PDO11: 1. B31...B30 (Augmented) set to 11b.  
2. B29..B28 (EPR Adjustable Voltage Supply) 01b  
3. B27..B26 (Peak Current) set to 00b  
4. B25..B17(maximum voltage) set to 48V  
5. B16(reserved) set to 0b  
6. B15..B8 (minimum voltage) set to 15V  
7. B7..B0 (PDP) set to 240W
7. The Tester check fails, and the test stops here if the UUT does not respond with the *EPR\_Request* Message [TEST.PD.EPR.SNK3.11#5]
  8. The Tester responds with the *Accept* Message followed by the *PS\_RDY* Message to the *EPR\_Request* Message and checking for *GoodCRC* [TEST.PD.EPR.SNK3.11#6] (establishing the EPR contract).
  9. The Tester sends the *PR\_Swap* Message
  10. The Tester checks that the UUT responds with [TEST.PD.EPR.SNK3.9#11]
    - a. *Not\_Supported* if the UUT is Consumer only

- b. Otherwise, *Reject*

*TEST.PD.EPR.SNK3.12 Negative Testing on EPR\_Request*

Description: The Tester as a Source checks that the UUT issues Hard Reset if the Tester rejects the first EPR\_Request in the EPR mode.

	Consumer Only	Provider Only	DRP, C/P, P/C
Rev3ChkdSnk	✓		✓

Check Applicability: The Test is applicable if the VIF Parameter EPR\_Supported\_As\_Snk is set to Yes

Test Procedures:

- There are 2 possible bring-up procedures:
  - The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8
  - The UUT has VIF field PD\_Port\_Type set to Consumer Only, Consumer /Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
- The Tester sends the *EPR\_Get\_Sink\_Cap* Message and performs below checks [TEST.PD.EPR.SNK3.12#2]
  - If the UUT is Consumer Only and VIF field EPR\_Supported\_as\_SNK is No, the Tester checks that the UUT responds with the *Not\_Supported* Message.
  - If the UUT is DRP or C/P or P/C and VIF field EPR\_Supported\_as\_SNK is No and VIF field EPR\_Supported\_as\_SRC is No, the Tester checks that the UUT responds with the *Not\_Supported* Message
  - If the UUT is not Consumer Only and VIF field EPR\_Supported\_as\_SNK is No and VIF field EPR\_Supported\_as\_SRC is Yes, the Tester checks that the UUT responds with the *Reject* or *Not\_Supported* Message
  - If the UUT VIF field EPR\_Supported\_as\_SNK is Yes, the Tester checks that the UUT responds with the *EPR\_Sink\_Capabilities* Message.

**NOTE:** *EPR\_Sink\_Capabilities* should not be the maximum power the UUT is able to request
- The Tester waits for 5 seconds and performs below checks, [TEST.PD.EPR.SNK3.12#3]
  - If the VIF field EPR\_Supported\_As\_Snk is No, then the check fails if the UUT sends an *EPR\_Mode* Message. The test stops here.

- b. If the VIF field `EPR_Supported_As_Snk` is Yes and the UUT does not send an `EPR_Mode` Enter message with the Action field set to 0x01, the test stops here, and the result is a warning.
5. If VIF field `Captive_Cable` = Yes, the Tester sends SOP' `Discover Identity` Command and checks the below fields in the response message `[TEST.PD.EPR.SNK3.12#4]`
  - a. `VBUS Current Handling Capability` set to 10b(5A)
  - b. `Maximum VBUS Voltage` set to 11b(50V)
  - c. `EPR Mode Capable` field set to 1b (EPR Mode Capable)
6. The Tester sends the `EPR_Mode` Message with the Action field set to 0x03(Enter\_Succeeded) and the Data field set to 0x0 within `tEnterEPR` from the `EPR_Mode` ( Enter) Message.
7. The Tester sends the `EPR_Source_Capabilities` Message that contains seven PDOs and four `EPR_PDOs` (240W)
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 0b if Tester is a Source only; set to 1b if Tester is a Dual-Role Power
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B27 (Unconstrained Power) set to 1b.
  - e. B26 (USB Communications Capable) set to 0b.
  - f. B25 (Dual-Role Data) set to 0b.
  - g. B24 (Unchunked Extended messages Supported) set to 0b or 1b depending on test condition.
  - h. B23 set to 1b (`EPR_Mode` Capable)
  - i. B22 (Reserved) set to 0b
  - j. B21...20 (Peak Current) set to 00b.
    - PDO1: 9. B19...10 (Voltage) set to 5V.  
10. B9...0 (Maximum Current) set to 3000mA
    - PDO2: 9. B19...10 (Voltage) set to 9V.  
10. B9...0 (Maximum Current) set to 3000mA
    - PDO3: 9. B19...10 (Voltage) set to 15V.  
10. B9...0 (Maximum Current) set to 3000mA
    - PDO4: 9. B19...10 (Voltage) set to 20V.  
10. B9...0 (Maximum Current) set to 5000mA
    - PDO5: B31..30 (APDO) set to 11b  
B29..28 (SPR AVS) set to 10b  
B27..26 (Peak Current) set to 00b  
B25..20 (Reserved) set to 00b  
B19..10 (9-15V current) set to 3000mA  
B9..0 (15-20v current) set to 5000mA
    - PDO6: All fields set to 0.

- PDO7: All fields set to 0.
  - PDO8: 9. B19...10 (Voltage) set to 28V.  
10. B9...0 (Maximum Current) set to 5000mA
  - PDO9: 9. B19...10 (Voltage) set to 36V.  
10. B9...0 (Maximum Current) set to 5000mA
  - PDO10: 9. B19...10 (Voltage) set to 48V.  
10. B9...0 (Maximum Current) set to 5000mA
  - PDO11: 1. B31...B30 (Augmented) set to 11b.  
2. B29..B28 (EPR Adjustable Voltage Supply) 01b  
3. B27..B26 (Peak Current) set to 00b  
4. B25..B17(maximum voltage) set to 48V  
5. B16(reserved) set to 0b  
6. B15..B8 (minimum voltage) set to 15V  
7. B7..B0 (PDP) set to 240W
- 8. The Tester check fails, and the test stops here if the UUT does not respond with the *EPR\_Request* Message [TEST.PD.EPR.SNK3.12#5]
- 9. The Tester sends the *Reject* Message.
- 10. The check fails and the test stops here if the UUT does not reply with the *Hard Reset* Signaling within *tProtErrHardReset*(this timing is measured from the last bit of the *GoodCRC* of the *EPR\_Request* Message to the last bit of the *Hard Reset* Signal) [TEST.PD.EPR.SNK3.12#6]
- 11. The Tester repeats steps **1 through 8**.
- 12. After the Tester sends the *Accept* Message, changes  $V_{BUS}$  voltage accordingly, and *PSRDY*, The Tester sends the *EPR\_Source\_Capabilities* Message that contains seven PDOs and 3 EPR\_PDOs **with changed capabilities**
  - a. B31...30 (Fixed Supply) set to 00b.
  - b. B29 (Dual-Role Power) set to 0b if Tester is a Source only; set to 1b if Tester is a Dual-Role Power
  - c. B28 (USB Suspend Supported) set to 0b.
  - d. B27 (Unconstrained Power) set to 1b.
  - e. B26 (USB Communications Capable) set to 0b.
  - f. B25 (Dual-Role Data) set to 0b.
  - g. B24 (Unchunked Extended messages Supported) set to 0b or 1b depending on test condition.
  - h. 23 set to 1b (*EPR\_Mode* Capable)
  - i. 22 (Reserved) set to 0b
  - j. B21...20 (Peak Current) set to 00b.
  - PDO1: 9. B19...10 (Voltage) set to 5V.  
10. B9...0 (Maximum Current) set to 3000mA



PDO2: 9. B19...10 (Voltage) set to 9V.  
 10. B9...0 (Maximum Current) set to 3000mA

PDO3: 9. B19...10 (Voltage) set to 15V.  
 10. B9...0 (Maximum Current) set to 3000mA

PDO4: 9. B19...10 (Voltage) set to 20V.  
 10. B9...0 (Maximum Current) set to 5000mA

PDO5: B31..30 (APDO) set to 11b  
 B29..28 (SPR AVS) set to 10b  
 B27..26 (Peak Current) set to 00b  
 B25..20 (Reserved) set to 00b  
 B19..10 (9-15V current) set to 3000mA  
 B9..0 (15-20v current) set to 5000mA

PDO6: All fields set to 0.

PDO7: All fields set to 0.

PDO8: 9. B19...10 (Voltage) set to 28V.  
 10. B9...0 (Maximum Current) set to 5000mA

PDO9: 9. B19...10 (Voltage) set to 36V.  
 10. B9...0 (Maximum Current) set to 5000mA

PDO10: 1. B31...B30 (Augmented) set to 11b.  
 2. B29..B28 (EPR Adjustable Voltage Supply) 01b  
 3. B27..B26 (Peak Current) set to 00b  
 4. B25..B17(maximum voltage) set to **36V**  
 5. B16(reserved) set to 0b  
 6. B15..B8 (minimum voltage) set to 15V  
 7. B7..B0 (PDP) set to 240W

13. The Tester fails the check, and the test stops here if the UUT does not respond with the *EPR\_Request* Message [TEST.PD.EPR.SNK3.12#7]
14. The Tester sends the *Reject* Message
15. The Tester checks that the UUT does not send Hard Reset signaling for at least *tProtErrHardReset* (this timing is measured from the last bit of the *GoodCRC* of the *EPR\_Request* Message)

## 8 Fast Role Swap Tests

### 8.1 Fast Role Swap Tests – Initial Source

#### 8.1.1 PD3 Mode

The tests in this subsection are run in PD3 mode.

The tests are based on the model of the UUT shown in **Figure 7** FRS Testing Concept.

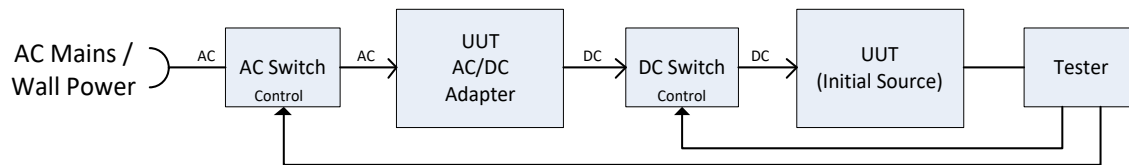


Figure 7 FRS Testing Concept

The assumption is that power to the Initial Source can be lost for either of two reasons:

1. The connection to the AC mains/wall power is broken (or the AC supply itself fails).
2. The connection from the UUT to its AC/DC Adapter is broken.

Either of these two events can be the cause of a Fast Role Swap request from the Initial Source. However, reason 2 is potentially more onerous on the UUT as the loss of usable power occurs with no chance of early warning. For some UUTs one or other of these interruptions may not be possible, so related testing need not take place.

During FRS testing, up to two switches, controlled by the tester (one AC and one DC), are connected up as part of the test setup. These are able to simulate the loss of power in the two distinct locations.

Please note that FRS tests apply whether or not FRS is supported, and the VIF parameter *Power\_Interruption\_Available* will be expected to be present in all cases.

The other factor which may affect the FRS performance is the voltage of the contract at the time of the power interruption. The main Initial Source test TEST.PD.FRS.SRC.1 will attempt to test both a 5V contract, and also the highest voltage contract achievable.

Test Conditions *Rev3ChkdSrc5VAC*, *Rev3ChkdSrc5VDC*, *Rev3ChkdSrcHiVAC* and *Rev3ChkdSrcHiVDC* are used to describe the combination of the position of the power interruption (AC or DC) and the contract voltage in force (see [Table 5](#)).

#### TEST.PD.FRS.SRC3.1 Normal Conditions

**Description:** Starting with the UUT as an initial Source, the Tester verifies that the UUT does Fast Role Swap correctly under normal conditions.

**Test Specific Tester Behavior:** N/A

**Test Conditions:**

This test is not applicable if VIF Parameter *Power\_Interruption\_Available* is *No Interruption Possible*. The *Rev3ChkdSrcEPRVDC* and *Rev3ChkdSrcEPRVAC* condition is not applicable if the UUT *EPR\_Supported\_As\_Src* is set to No.

	Consumer Only, Provider Only,	C/P, P/C, DRP
<i>Rev3ChkdSrc5VAC</i>		✓
<i>Rev3ChkdSrc5VDC</i>		✓
<i>Rev3ChkdSrcHiVAC</i>		✓
<i>Rev3ChkdSrcHiVDC</i>		✓
<i>Rev3ChkdSrcEPRVDC</i>		✓
<i>Rev3ChkdSrcEPRVAC</i>		✓

**Test Procedures:**

1. There are 2 possible bring-up procedures:

- a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, or DRP, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. For Test Conditions *Rev3ChkdSrcHiVAC* or *Rev3ChkdSrcHiVDC*: The Tester makes a Request for the the highest fixed PDO number and checks that the contract is made. [TEST.PD.FRS.SRC3.1#1]
3. The Tester sends *Get\_Sink\_Cap* to the UUT
4. If a *Sink\_Capabilities* Message is not received, the test ends here, and the check fails. [TEST.PD.FRS.SRC3.1#2]
5. For Test Conditions *Rev3ChkdSrcEPRVAC* or *Rev3ChkdSrcEPRVDC*:
  - a. If the *EPR Mode* Capable bit is not set to 1 in the *Source\_Capabilities* Message, then test condition stops here and results in a warning
  - b. The Tester sends *Request* with the *EPR Mode* Capable bit set to 1.
  - c. The Tester sends the *EPR Mode* Entry Message and follows EPR Entry sequence and sends the *EPR\_Request* Message with the highest fixed PDO number and checks that the contract is made. [TEST.PD.FRS.SRC3.1#12]
6. The Tester operates the appropriate power disconnection switch (see Appendix B) depending on the Test Condition.
7. The Tester verifies the following: [TEST.PD.FRS.SRC3.1#3]
  - a. If Fast Role Swap USB Type-C Current field is 00b (in the *Sink\_Capabilities* Message), check that no Fast Role Swap signal is received. The test ends here.
  - b. If Fast Role Swap USB Type-C Current field is not 00b (in the *Sink\_Capabilities* Message), check that Fast Role Swap signal is received within 60 seconds after power disconnection. Also, check that the Fast Role Swap signal meets *tFRSwapTx*. The receiver detection level used to detect the Fast Role Swap signal shall be the mid-range value of *vFrSwapCableTx* (i.e.520mV). For Test Conditions *Rev3ChkdSrcHiVAC* or *Rev3ChkdSrcHiVDC*: The check fails if the FRS Signal starts after *tFRSwapRx* max before the time  $V_{BUS}$  falls to 90% of the nominal value.
8. After receiving the Fast Role Swap signal, the  $V_{BUS}$  electrical and PD messaging test sub-steps are performed separately. In the following sub-steps, the sequences of the  $V_{BUS}$  electrical behavior and PD messaging are only bounded at the operating point referred to as *Stage 1*.
  - a.  $V_{BUS}$  electrical sub-steps:
    - i. The Tester immediately stops sinking current through  $V_{BUS}$  after detecting Fast Role Swap signal
    - ii. The Tester waits for  $V_{BUS}$  to be below *vSafe5V* max and this operating point is referred to as Stage 1
    - iii. The Tester applies *vSafe5V* on  $V_{BUS}$  at *tSrcFRSwap* max after receiving the Fast Role Swap signal and  $V_{BUS}$  has dropped below *vSafe5V* min. After the start of Fast Role Swap signal, the Tester checks that the UUT does not draw more than *iSnkStdby*, either while  $V_{BUS}$  is below *vSafe5V*

- (min) or *tSnkFRSwap* has elapsed, whichever comes first.  
[TEST.PD.FRS.SRC3.1#4]
- iv. After *tSnkFRSwap*, check that the UUT does not draw more current than it declared it would in the Fast Role Swap USB Type-C Current field  
[TEST.PD.FRS.SRC3.1#5].
- b. PD messaging sub-steps:
- i. The Tester checks that the UUT presents *SinkTxOk* within 1 millisecond after receiving Fast Role Swap signal [TEST.PD.FRS.SRC3.1#6].
  - ii. The Tester sends an *FR\_Swap* Message at *tFRSwapInit* max after receiving the Fast Role Swap signal. If the Tester cannot send the *FR\_Swap* Message because it cannot start an AMS, the check fails [TEST.PD.FRS.SRC3.1#7].
  - iii. Check that the UUT sends an *Accept* Message within *tReceiverResponse* max from the last bit of the EOP of the *FR\_Swap* Message, and check that the Port Power Role field = 1b (Source) in this message [TEST.PD.FRS.SRC3.1#8].
  - iv. Check that the UUT presents Rd after the operating point has reached Stage 1, and after the UUT has sent the *Accept* Message [TEST.PD.FRS.SRC3.1#9].
  - v. Check that the UUT sends a *PS\_RDY* Message after presenting Rd, and within *tFRSwap5V* of Stage 1 or of the last bit of the EOP of GoodCRC corresponding to the *Accept* Message if that is later. Check that the Port Power Role field = 0b (Sink) in the *PS\_RDY* Message.  
[TEST.PD.FRS.SRC3.1#10]
  - vi. The Tester changes the presented CC-line termination from Rd to Rp at a time *tFRSwapComplete* minus 1 millisecond after receiving the last bit of the EOP of the *PS\_RDY* Message. The Rp value the Tester presents shall match *FR\_Swap\_Reqd\_Type\_C\_Current\_As\_Initial\_Source* parameter as specified in the VIF.
  - vii. The Tester starts sending a *PS\_RDY* Message at a time *tFRSwapComplete* minus 1 millisecond after the Tester is providing power.
  - viii. As the new Source, the Tester sends a *Source\_Capabilities* Message that matches the UUT sink requirements and checks that the UUT follows COMMON.PROC.PD.10 to establish a new contract  
[TEST.PD.FRS.SRC3.1#11].

9. Repeat the test for every test condition based on *Power\_Interruption\_Available VIF* field.

#### TEST.PD.FRS.SRC3.2 Provider Only Checks

Description: Starting with the UUT as an initial Source, the Tester verifies that the UUT does not attempt Fast Role Swap under any conditions.

Test Specific Tester Behavior: N/A

Test Conditions:

This test is not applicable if VIF Parameter *Power\_Interruption\_Available* is *No Interruption Possible*.

If VIF Parameter *Power\_Interruption\_Available* is *AC mains/wall power Only* or *AC mains/wall power and DC Power* then the first test only shall be run, otherwise the second test only shall be run.

	Consumer Only C/P, P/C, DRP	Provider Only
<i>Rev3ChkdSrc5VAC</i>		✓
<i>Rev3ChkdSrc5VDC</i>		✓

#### Test Procedures:

1. The Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester sends *Get\_Sink\_Cap* to the UUT
3. Check that a *Not\_Supported* Message is received. [TEST.PD.FRS.SRC3.2#1]
4. The Tester operates the appropriate power disconnection switch (see Appendix B) depending on the Test Condition.
5. Check that no Fast Role Swap signal is received. The test ends here. [TEST.PD.FRS.SRC3.2#2]

#### TEST.PD.FRS.SRC3.3 *GoodCRC* Not Sent In Response To Accept

Description: Starting with the UUT as an initial Source, after the receipt of the *Accept* Message for *FR\_Swap*, the Tester intentionally does not reply *GoodCRC* in order to force a *Hard Reset* on the UUT and verifies it is correctly implemented.

#### Test Conditions:

This test is not applicable if VIF parameter *FR\_Swap\_Reqd\_Type\_C\_Current\_As\_Initial\_Source* is 00b, or VIF Parameter *Power\_Interruption\_Available* is *No Interruption Possible*.

If VIF Parameter *Power\_Interruption\_Available* is *AC mains/wall power Only* or *AC mains/wall power and DC Power* then the first test only shall be run, otherwise the second test only shall be run.

	Consumer Only, Provider Only,	C/P, P/C, DRP
<i>Rev3ChkdSrc5VAC</i>		✓
<i>Rev3ChkdSrc5VDC</i>		✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider, the Tester runs bring-up

- b. procedure with the UUT as a Source COMMON.PROC.BU.7. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, or DRP, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester operates the appropriate power disconnection switch (see Appendix B) depending on the Test Condition. If the Fast Swap Signal is not detected within 60 second time period, the test stops here.
3. After receiving the Fast Role Swap signal, the  $V_{BUS}$  electrical and PD messaging test sub-steps are performed separately.
  - a.  $V_{BUS}$  electrical sub-steps:
    - i. The Tester immediately stops sinking current through  $V_{BUS}$  after detecting Fast Role Swap signal
    - ii. The Tester waits for  $V_{BUS}$  to be below  $vSafe5V$  max
    - iii. The Tester applies  $vSafe5V$  on  $V_{BUS}$  at  $tSrcFRSwap$  max after receiving the Fast Role Swap signal and  $V_{BUS}$  has dropped below  $vSafe5V$  min.
  - b. PD messaging sub-steps:
    - i. The Tester sends an *FR\_Swap* Message. If the Tester cannot send the *FR\_Swap* Message because it cannot start an AMS, the check fails, and the test ends here [TEST.PD.FRS.SRC3.3#1].
    - ii. Upon receipt of the *Accept* Message, the Tester intentionally does not reply with a *GoodCRC* Message.
    - iii. After nRetryCount retries of the *Accept* Message, the check fails if *Hard Reset* is not detected within  $tReceive$  max +  $tProtErrHardReset$  max, the delay is measured from the time the last bit of the last retransmitted *Accept* Message EOP has been received to the time the first bit of the *Hard Reset* is received [TEST.PD.FRS.SRC3.3#2].

#### TEST.PD.FRS.SRC3.4 *GoodCRC* Not Sent In Response To *PS\_RDY*

**Description:** Starting with the UUT as an initial Source, after the receipt of the *PS\_RDY* Message during the Fast Role Swap process, the Tester intentionally does not reply *GoodCRC* in order to force an Error Recovery on the UUT and verifies it is correctly implemented.

#### **Test Conditions:**

This test is not applicable if VIF parameter *FR\_Swap\_Reqd\_Type\_C\_Current\_As\_Initial\_Source* is 00b, or VIF Parameter *Power\_Interruption\_Available* is *No Interruption Possible*.

If VIF Parameter *Power\_Interruption\_Available* is *AC mains/wall power Only* or *AC mains/wall power and DC Power* then the first test only shall be run, otherwise the second test only shall be run.

	Consumer Only, Provider Only,	C/P, P/C, DRP
<i>Rev3ChkdSrc5VAC</i>		✓
<i>Rev3ChkdSrc5VDC</i>		✓

## Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, or DRP, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester operates the appropriate power disconnection switch (see Appendix B) depending on the Test Condition. If the Fast Swap Signal is not detected within 60 second time period, the test stops here.
3. After receiving the Fast Role Swap signal, the V<sub>BUS</sub> electrical and PD messaging test sub-steps are performed separately.
  - b. V<sub>BUS</sub> electrical sub-steps:
    - iv. The Tester immediately stops sinking current through V<sub>BUS</sub> after detecting Fast Role Swap signal.
    - v. The Tester waits for V<sub>BUS</sub> to be below *vSafe5V* max
    - vi. The Tester applies *vSafe5V* on V<sub>BUS</sub> at *tSrcFRSwap* max after receiving the Fast Role Swap signal and V<sub>BUS</sub> has dropped below *vSafe5V* min.
  - a. PD messaging sub-steps:
    - i. The Tester sends a *FR\_Swap* Message. If the Tester cannot send the *FR\_Swap* Message because it cannot start an AMS, the check fails, and the test ends here [TEST.PD.FRS.SRC3.4#1].
    - ii. Upon receipt of the *Accept* Message, the Tester replies with a *GoodCRC* message.
    - iii. The check fails and the test ends here if *PS\_RDY* is not received within *tPSSourceOff* min. The delay is measured from the time the last bit of the EOP of the *GoodCRC* Message sent by the Tester in response to the *Accept* Message to the time the last bit of the EOP of the *GoodCRC* Message corresponding to the received the *PS\_RDY* Message. [TEST.PD.FRS.SRC3.4#2]
    - iv. Upon receipt of the *PS\_RDY* Message, the Tester intentionally does not reply with a *GoodCRC* Message.
    - v. The check fails if a USB Type-C Error Recovery is not detected within *tCtsErrorRecovery* after the last bit of the EOP of the last retried *PS\_RDY* Message. [TEST.PD.FRS.SRC3.4#3]

### TEST.PD.FRS.SRC3.5 PSSourceOnTimer Deadline

Description: Starting with the UUT as an initial Source, after a *FR\_Swap* accepted, the Tester sends *PS\_RDY* at the deadline limit of *tPSSourceOn* min and verifies the UUT accepts this message.

### Test Conditions:



This test is not applicable if VIF parameter *FR\_Swap\_Reqd\_Type\_C\_Current\_As\_Initial\_Source* is 00b, or VIF Parameter *Power\_Interruption\_Available* is *No Interruption Possible*.  
If VIF Parameter *Power\_Interruption\_Available* is *AC mains/wall power Only* or *AC mains/wall power and DC Power* then the first test only shall be run, otherwise the second test only shall be run.

	Consumer Only, Provider Only,	C/P, P/C, DRP
<i>Rev3ChkdSrc5VAC</i>		✓
<i>Rev3ChkdSrc5VDC</i>		✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field *PD\_Port\_Type* set to Provider/Consumer, or DRP, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester operates the appropriate power disconnection switch (see Appendix B) depending on the Test Condition. If the Fast Swap Signal is not detected within 60 second time period, the test stops here.
3. After receiving the Fast Role Swap signal, the  $V_{BUS}$  electrical and PD messaging test sub-steps are performed separately.
  - a.  $V_{BUS}$  electrical sub-steps:
    - vii. The Tester immediately stops sinking current through  $V_{BUS}$  after detecting Fast Role Swap signal
    - viii. The Tester waits for  $V_{BUS}$  to be below *vSafe5V* max
    - ix. The Tester applies *vSafe5V* on  $V_{BUS}$  at *tSrcFRSwap* max after receiving the Fast Role Swap signal and  $V_{BUS}$  has dropped below *vSafe5V* min.
  - b. PD messaging sub-steps:
    - i. The Tester sends a *FR\_Swap* Message. If the Tester cannot send the *FR\_Swap* Message because it cannot start an AMS, the check fails, and the test ends here. [TEST.PD.FRS.SRC3.5#1]
    - ii. Upon receipt of the *Accept* Message, the Tester replies with a *GoodCRC* Message.
    - iii. Upon receipt of the *PS\_RDY* Message, the Tester replies with a *GoodCRC* Message.
    - iv. The Tester sends the *PS\_RDY* Message between *tCtsPSSourceOnDeadline* min and max after transmitting the last bit of the *GoodCRC* Message EOP, corresponding to the received *PS\_RDY* Message.
    - v. The test passes if a *GoodCRC* is received. [TEST.PD.FRS.SRC3.5#2]

- vi. The test fails if Error Recovery is detected when waiting the *GoodCRC* Message or before sending the *PS\_RDY*. [TEST.PD.FRS.SRC3.5#3]

#### TEST.PD.FRS.SRC3.6 PSSourceOnTimer Timeout

**Description:** Starting with the UUT as an initial Source, after a *FR\_Swap* accepted, the Tester intentionally does not send *PS\_RDY*, in order to force a PSSourceOnTimer timeout on the UUT and verifies it is correctly implemented.

#### Test Conditions:

This test is not applicable if VIF parameter *FR\_Swap\_Reqd\_Type\_C\_Current\_As\_Initial\_Source* is 00b, or VIF Parameter *Power\_Interruption\_Available* is *No Interruption Possible*.

If VIF Parameter *Power\_Interruption\_Available* is *AC mains/wall power Only* or *AC mains/wall power and DC Power* then the first test only shall be run, otherwise the second test only shall be run.

	Consumer Only, Provider Only,	C/P, P/C, DRP
<i>Rev3ChkdSrc5VAC</i>		✓
<i>Rev3ChkdSrc5VDC</i>		✓

#### Test Procedures:

1. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.7.
  - b. The UUT has VIF field *PD\_Port\_Type* set to Provider/Consumer, or DRP, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
2. The Tester operates the appropriate power disconnection switch (see Appendix B) depending on the Test Condition.If the Fast Swap Signal is not detected within 60 second time period, the test stops here.
3. After receiving the Fast Role Swap signal, the  $V_{BUS}$  electrical and PD messaging test sub-steps are performed separately.
  - a.  $V_{BUS}$  electrical sub-steps:
    - i. The Tester immediately stops sinking current through  $V_{BUS}$  after detecting Fast Role Swap signal.
    - ii. The Tester waits for  $V_{BUS}$  to be below *vSafe5V* max
    - iii. The Tester applies *vSafe5V* on  $V_{BUS}$  at *tSrcFRSwap* max after receiving the Fast Role Swap signal and  $V_{BUS}$  has dropped below *vSafe5V* min.
  - b. PD messaging sub-steps:
    - i. The Tester sends a *FR\_Swap* Message. If the Tester cannot send the *FR\_Swap* Message because it cannot start an AMS, the check fails, and the test ends here. [TEST.PD.FRS.SRC3.6#1]

- ii. Upon receipt of the *Accept* Message, the Tester replies with a *GoodCRC* Message.
- iii. Upon receipt of the *PS\_RDY* Message, the Tester replies with a *GoodCRC* Message.
- iv. The Tester intentionally does not send *PS\_RDY*, in order to force a *PSSourceOnTimer* timeout on the UUT.
- v. Check that Error Recovery is detected in the range *tPSSourceOn* min to *tPSSourceOn* max after the time the last bit of the EOP of the *GoodCRC* Message corresponding to the received *PS\_RDY* Message was sent.  
[TEST.PD.FRS.SRC.6#2]

## 8.2 Fast Role Swap Tests – Initial Sink

### 8.2.1 PD3 Mode

The tests in this subsection are run in PD3 mode.

#### TEST.PD.FRS.SNK3.1 Normal Conditions

**Description:** Starting with the UUT as an initial Sink, the Tester verifies that the UUT does Fast Role Swap correctly under normal conditions.

#### Test Conditions:

The Rev3ChkdSnkHiv condition is not applicable if the UUT has only Sink PDO

The Rev3ChkdSnkEPR condition is not applicable if the UUT EPR\_Supported\_As\_Snk is set to No.

	P/C, C/P, DRP	Consumer Only, Provider Only
<i>Rev3ChkdSnk5V</i>	✓	
<i>Rev3ChkdSnkHiV</i>	✓	
<i>Rev3ChkdSnkEPR</i>	✓	

#### Test Procedures:

1. For Test Condition *Rev3ChkdSnkHiV*, during step 3 the Tester sends a *Source\_Capabilities* Message with two PDOs:
  - a. a Fixed Supply PDO at 5V/0A, and
  - b. a Fixed Supply PDO that matches the UUT's Sink PDO with the highest voltage.
  - c. If UUT does not select the second PDO, the test stops here, and the result is a warning.
2. For Test Condition *Rev3ChkdSnkEPR*, during step 3 the Tester sends a *Source\_Capabilities* Message with EPR Mode Capable bit set to 1.
  - a. During EPR Mode Entry, the Tester advertises an EPR Fixed Supply PDO that matches the UUT's Sink PDO with the highest voltage. If the UUT does not enter the EPR Mode, the test stops here and the results in a warning.
  - b. In the EPR Mode the Tester sends the Fast Role Swap signal as described in Step 5.

3. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8 (Dual-Role Power bit B29 in the single PDO is set to 1b)
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2 (Dual-Role Power bit B29 in the single PDO is set to 1b)
4. Tester waits *tCtsAfterLastPoint* to allow UUT to request Sink Caps.
  - a. If it does, the Tester replies with a single PDO set to what the VIF parameter FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink says the UUT will support as a Source (i.e. 5V @ 500mA, 1.5A or 3A).
  - b. If no Get Sink Caps message received, then check that the VIF parameter FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink is 00b.  
[TEST.PD.FRS.SNK3.1#1]
5. The Tester sends the Fast Role Swap signal to the UUT, with a duration of *tFRSwapTx* min. The signal is applied via a resistance of *rFRSwapTx* (5Ω) to ground.
6. After sending the Fast Role Swap signal, the V<sub>BUS</sub> electrical and PD messaging test sub-steps are performed separately.
  - a. V<sub>BUS</sub> electrical sub-steps:
    - i. The Tester immediately turns off the V<sub>BUS</sub> at the end of Fast Role Swap signal. The Tester as a Source shall present a capacitance of 10uF on the V<sub>BUS</sub> line.
    - ii. The Tester draws no current until *tSnkFRSwap* has passed, since the start of the Fast Role Swap signal or since when the V<sub>BUS</sub> falls below *vSafe5V* min, whichever comes later. After that, the Tester draws the current specified in VIF parameter FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink. After the Tester has turned off V<sub>BUS</sub>, the operating point when V<sub>BUS</sub> is below *vSafe5V* max is referred to as *Stage 1*.
    - iii. The Tester makes the following check on V<sub>BUS</sub>:
      1. If the VIF parameter FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink is 00b, the Tester checks that the UUT does not supply V<sub>BUS</sub> within 1s. This branch of the test ends here. [TEST.PD.FRS.SNK3.1#8]
      2. In any other case, it verifies that the UUT as a new Source supplies V<sub>BUS</sub> and it reaches *vSafe5V* min within *tSrcFRSwap* of both the Fast Role Swap signal having had time to be detected, and V<sub>BUS</sub> being below *vSafe5V* min (due to Tester turning off V<sub>BUS</sub>).  
[TEST.PD.FRS.SNK3.1#2]
  - b. PD messaging sub-steps:
    - i. The Tester presents *SinkTxOk* immediately after sending Fast Role Swap signal.
    - ii. The Tester makes the following message check:

1. If the VIF parameter *FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink* is 00b, the Tester checks that the UUT does not send an *FR\_Swap* Message. [TEST.PD.FRS.SNK3.1#3] This branch of the test ends here.
2. In any other case, it checks that *FR\_Swap* Message starts to be sent by the UUT within *tFRSwapRx* max plus *tFRSwapInit* max of the start of the Fast Role Swap signal. [TEST.PD.FRS.SNK3.1#4]
- iii. The Tester sends an *Accept* Message at the boundary of *tReceiverResponse* max from the last bit of the EOP of the *FR\_Swap* Message
- iv. The Tester changes from presenting Rp to Rd as soon as the operating point has reached *Stage 1*.
- v. The Tester sends *PS\_RDY* Message immediately after changing from presenting Rp to Rd (in sub-step d)
- vi. The Tester checks that UUT changes from presenting Rd to Rp within *tFRSwapComplete* of having sent *PS\_RDY* Message (in sub-step e) [TEST.PD.FRS.SNK3.1#5]
- vii. The Tester checks that it starts receiving *PS\_RDY* Message from the UUT within *tFRSwapComplete* of having sent *PS\_RDY* Message (in sub-step e). [TEST.PD.FRS.SNK3.1#6]
- viii. The Tester checks that the UUT sends a *Source\_Capabilities* Message after *tSwapSourceStart* and before *tFirstSourceCap* max. [TEST.PD.FRS.SNK3.1#7]

#### TEST.PD.FRS.SNK3.2 Normal Conditions, Consumer Only

**Description:** Starting with the UUT as an initial Sink, the Tester verifies that the Consumer Only UUT correctly ignores Fast Role Swap.

##### Test Conditions:

	Consumer Only	Provider Only, P/C, C/P, DRP
<i>Rev3ChkdSnk</i>	✓	

##### Test Procedures:

1. The Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2 (Dual-Role Power bit B29 in the single PDO is set to 1b)
2. Tester waits *tCtsAfterLastPoint* to allow UUT the opportunity to request Sink Caps.
3. Check that the UUT does not send a Get Sink Caps message. [TEST.PD.FRS.SNK3.2#1]
4. The Tester sends a Fast Role Swap signal to the UUT, with a duration of *tFRSwapTx* min. The signal is applied via a resistance of *rFRSwapTx* (5Ω) to ground.
5. After sending the Fast Role Swap signal, the V<sub>BUS</sub> electrical and PD messaging test sub-steps are performed separately.
  - a. V<sub>BUS</sub> electrical sub-steps:

- i. The Tester immediately turns off the  $V_{BUS}$  at the end of Fast Role Swap signal. The Tester as a Source shall present a capacitance of 10uF on the  $V_{BUS}$  line.
  - ii. The Tester draws no current until  $t_{SnkFRSwap}$  has passed, since the start of the Fast Role Swap signal or since when the  $V_{BUS}$  falls below  $v_{Safe5V}$  min, whichever comes later. After that, the Tester attempts to draw 100mA. (This is necessary to detect whether  $V_{BUS}$  is being applied by UUT.)
  - iii. The Tester checks that the UUT does not supply  $V_{BUS}$  within 1s of the Tester  $V_{BUS}$  falling below  $v_{Safe5V}$  min. [TEST.PD.FRS.SNK3.2#2]
- b. PD messaging sub-steps:
  - i. The Tester presents *SinkTxOk* immediately after sending Fast Role Swap signal.
  - ii. The Tester checks that the UUT does not send an *FR\_Swap* Message within 1s. [TEST.PD.FRS.SNK3.2#3]

#### TEST.PD.FRS.SNK3.3 *FR\_Swap* Not Sent

**Description:** Starting with the UUT as an initial Sink, after the receipt of the *FR\_Swap* Message, the Tester intentionally does not reply *GoodCRC* in order to force an Error Recovery on the UUT and verifies it is correctly implemented.

#### Test Conditions:

	P/C, C/P, DRP	Consumer Only, Provider Only
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. If VIF field *FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink* is set in the vendor file to 00b, this test passes and stops here.
2. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8 (Dual-Role Power bit B29 in the single PDO is set to 1b)
  - b. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2 (Dual-Role Power bit B29 in the single PDO is set to 1b)
3. The Tester sends a Fast Role Swap signal to the UUT, with a duration of  $t_{FRSwapTx}$  min.
4. After sending the Fast Role Swap signal, the  $V_{BUS}$  electrical and PD messaging test sub-steps are performed separately.
  - a.  $V_{BUS}$  electrical sub-steps:

- i. The Tester immediately turns off the  $V_{BUS}$  at the end of Fast Role Swap signal. The Tester as a Source shall present a capacitance of 10uF on the  $V_{BUS}$  line.
  - ii. The Tester draws no current until  $t_{SnkFRSwap}$  has passed, since the start of the Fast Role Swap signal or since when the  $V_{BUS}$  falls below  $v_{Safe5V}$  min, whichever comes later. After that, the Tester draws the current specified in VIF parameter  $FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink$ .
- b. PD messaging sub-steps:
  - i. The Tester presents *SinkTxOk* immediately after sending Fast Role Swap signal.
  - ii. The Tester checks that *FR\_Swap* Message starts to be sent by the UUT within  $t_{FRSwapRx}$  max plus  $t_{FRSwapInit}$  max of the start of the Fast Role Swap signal. [TEST.PD.FRS.SNK3.3#1]
  - iii. Upon receipt of the *FR\_Swap* Message, the Tester intentionally does not reply with a *GoodCRC* Message
  - iv. The check fails if Error Recovery is not detected within  $t_{CtsErrorRecovery}$  after the last bit of the EOP of the last retried *FR\_Swap* Message. [TEST.PD.FRS.SNK3.3#2]

#### TEST.PD.FRS.SNK3.4 *SendResponseTimer Timeout*

**Description:** Starting with the UUT as an initial Sink, after the receipt of the *FR\_Swap* Message, the Tester intentionally does not reply with *Accept* in order to force a *SenderResponseTimer* time out on the UUT and verifies it is correctly implemented.

#### Test Conditions:

	P/C, C/P, DRP	Consumer Only, Provider Only
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. If VIF field  $FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink$  is set in the vendor file to 00b, this test passes and stops here.
2. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field  $PD\_Port\_Type$  set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8 (Dual-Role Power bit B29 in the single PDO is set to 1b)
  - b. The UUT has VIF field  $PD\_Port\_Type$  set to Consumer/Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2 (Dual-Role Power bit B29 in the single PDO is set to 1b)



3. The Tester sends a Fast Role Swap signal to the UUT, with a duration of *tFRSwapTx* min.
4. After sending the Fast Role Swap signal, the V<sub>BUS</sub> electrical and PD messaging test sub-steps are performed separately.
  - a. V<sub>BUS</sub> electrical sub-steps:
    - i. The Tester immediately turns off the V<sub>BUS</sub> at the end of Fast Role Swap signal. The Tester as a Source shall present a capacitance of 10uF on the V<sub>BUS</sub> line.
    - ii. The Tester draws no current until *tSnkFRSwap* has passed, since the start of the Fast Role Swap signal or since when the V<sub>BUS</sub> falls below *vSafe5V* min, whichever comes later. After that, the Tester draws the current specified in VIF parameter FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink.
  - b. PD messaging sub-steps:
    - i. The Tester presents *SinkTxOk* immediately after sending Fast Role Swap signal.
    - ii. The Tester checks that *FR\_Swap* Message starts to be sent by the UUT within *tFRSwapRx* max plus *tFRSwapInit* max of the start of the Fast Role Swap signal. [TEST.PD.FRS.SNK3.4#1]
    - iii. Upon receipt of the *FR\_Swap* Message, the Tester replies with a *GoodCRC* Message but intentionally does not reply with *Accept* Message to the UUT.
    - iv. Check that Error Recovery is detected between *tSenderResponse* min and *tSenderResponse* max + *InitiateErrorRecovery* (see Table 19 Timing Table & Calculations). The delay is measured from the time the last bit of the *GoodCRC* Message EOP has been sent to the time the Error Recovery starts. [TEST.PD.FRS.SNK3.4#2]

#### TEST.PD.FRS.SNK3.5 PSSourceOffTimer Deadline

**Description:** Starting with the UUT as an initial Sink, after accepting a *FR\_Swap*, the Tester sends *PS\_RDY* at the deadline limit of tPSSourceOff min and verifies the UUT accepts this message.

**Test Conditions:**

	P/C, C/P, DRP	Consumer Only, Provider Only
<i>Rev3ChkdSnk</i>	✓	

**Test Procedures:**

1. If VIF field FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink is set in the vendor file to 00b, this test passes and stops here.
2. There are 2 possible bring-up procedures:

- a. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8 (Dual-Role Power bit B29 in the single PDO is set to 1b)
  - b. The UUT has VIF field PD\_Port\_Type set to Consumer/Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2 (Dual-Role Power bit B29 in the single PDO is set to 1b)
3. The Tester sends a Fast Role Swap signal to the UUT, with a duration of *tFRSwapTx* min.
4. After sending the Fast Role Swap signal, the V<sub>BUS</sub> electrical and PD messaging test sub-steps are performed separately.
  - a. V<sub>BUS</sub> electrical sub-steps:
    - i. The Tester immediately turns off the V<sub>BUS</sub> at the end of Fast Role Swap signal. The Tester as a Source shall present a capacitance of 10uF on the V<sub>BUS</sub> line.
    - ii. The Tester draws no current until *tSnkFRSwap* has passed, since the start of the Fast Role Swap signal or since when the V<sub>BUS</sub> falls below *vSafe5V* min, whichever comes later. After that, the Tester draws the current specified in VIF parameter FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink. After the Tester has turned off V<sub>BUS</sub>, the operating point when V<sub>BUS</sub> is below *vSafe5V* max is referred to as *Stage 1*.
  - b. PD messaging sub-steps:
    - i. The Tester presents *SinkTxOk* immediately after sending Fast Role Swap signal.
    - ii. The Tester checks that the *FR\_Swap* Message starts to be sent by the UUT within *tFRSwapRx* max plus *tFRSwapInit* max of the start of the Fast Role Swap signal. [TEST.PD.FRS.SNK3.5#1]
    - iii. Upon receipt of the *FR\_Swap* Message, the Tester replies with a *GoodCRC* and sends an *Accept* Message to the UUT.
    - iv. The Tester changes from presenting Rp to Rd as soon as the operating point has reached Stage 1.
    - v. The Tester sends *PS\_RDY* between *tCtsPSSourceOffDeadline* min and max after receiving the last bit of the *GoodCRC* Message EOP corresponding to the transmitted the *Accept* Message.
    - vi. The check fails if Error Recovery is detected when waiting the *GoodCRC* Message or before the *PS\_RDY* is sent. [TEST.PD.FRS.SNK3.5#2]

#### TEST.PD.FRS.SNK3.6 PSSourceOffTimer Timeout

**Description:** Starting with the UUT as an initial Sink, after accepting a *FR\_Swap*, the Tester intentionally does not send *PS\_RDY*, to force a PSSourceOffTimer timeout on the UUT and verifies it is correctly implemented.

**Test Conditions:**

P/C, C/P, DRP	Consumer Only, Provider Only
------------------	---------------------------------

Test Procedures:

1. If VIF field *FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink* is set in the vendor file to 00b, this test passes and stops here.
2. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8 (Dual-Role Power bit B29 in the single PDO is set to 1b)
  - b. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2 (Dual-Role Power bit B29 in the single PDO is set to 1b)
3. The Tester sends a Fast Role Swap signal to the UUT, with a duration of *tFRSwapTx* min.
4. After sending the Fast Role Swap signal, the  $V_{BUS}$  electrical and PD messaging test sub-steps are performed separately.
  - a.  $V_{BUS}$  electrical sub-steps:
    - i. The Tester immediately turns off the  $V_{BUS}$  at the end of Fast Role Swap signal. The Tester as a Source shall present a capacitance of 10uF on the  $V_{BUS}$  line.
    - ii. The Tester draws no current until *tSnkFRSwap* has passed, since the start of the Fast Role Swap signal or since when the  $V_{BUS}$  falls below *vSafe5V* min, whichever comes later. After that, the Tester draws the current specified in VIF parameter *FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink*. After the Tester has turned off  $V_{BUS}$ , the operating point when  $V_{BUS}$  is below *vSafe5V* max is referred to as *Stage 1*.
  - b. PD messaging sub-steps:
    - i. The Tester presents *SinkTxOk* immediately after sending Fast Role Swap signal.
    - ii. The Tester checks that *FR\_Swap* Message starts to be sent by the UUT within *tFRSwapRx* max plus *tFRSwapInit* max of the start of the Fast Role Swap signal. [TEST.PD.FRS.SNK3.6#1]
    - iii. Upon receipt of the *FR\_Swap* Message, the Tester replies with a *GoodCRC* and sends an *Accept* Message to the UUT.
    - iv. The Tester continues to assert *Rp* (i.e. the Tester does not assert *Rd* so that the Tester can detect the UUT entering Error Recovery).
    - v. The Tester intentionally does not send *PS\_RDY*, to force a *PSSourceOffTimer* timeout on the UUT.
    - vi. Check that Error Recovery is detected between *tPSSourceOff* min, and *tPSSourceOff* max. The delay is measured from the time the last bit of the

*GoodCRC* Message EOP corresponding to the *Accept* Message has been sent to the time the Error Recovery starts. [TEST.PD.FRS.SNK3.6#1]

#### TEST.PD.FRS.SNK3.7 *GoodCRC Not Sent in Response to PS\_RDY*

Description: Starting with the UUT as an initial Sink, after the receipt of the *PS\_RDY* Message, the Tester intentionally does not reply *GoodCRC* in order to force an Error Recovery on the UUT and verifies it is correctly implemented.

#### Test Conditions:

	P/C, C/P, DRP	Consumer Only, Provider Only
<i>Rev3ChkdSnk</i>	✓	

#### Test Procedures:

1. If VIF field *FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink* is set in the vendor file to 00b, this test passes and stops here.
2. There are 2 possible bring-up procedures:
  - a. The UUT has VIF field *PD\_Port\_Type* set to Provider/Consumer, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.8 (Dual-Role Power bit B29 in the single PDO is set to 1b)
  - b. The UUT has VIF field *PD\_Port\_Type* set to Consumer/Provider or DRP, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2 (Dual-Role Power bit B29 in the single PDO is set to 1b)
3. The Tester sends a Fast Role Swap signal to the UUT, with a duration of *tFRSwapTx* min.
4. After sending the Fast Role Swap signal, the *V<sub>BUS</sub>* electrical and PD messaging test sub-steps are performed separately.
  - a. *V<sub>BUS</sub>* electrical sub-steps:
    - i. The Tester immediately turns off the *V<sub>BUS</sub>* at the end of Fast Role Swap signal. The Tester as a Source shall present a capacitance of 10uF on the *V<sub>BUS</sub>* line.
    - ii. The Tester draws no current until *tSnkFRSwap* has passed, since the start of the Fast Role Swap signal or since when the *V<sub>BUS</sub>* falls below *vSafe5V* min, whichever comes later. After that, the Tester draws the current specified in VIF parameter *FR\_Swap\_Type\_C\_Current\_Capability\_As\_Initial\_Sink*. After the Tester has turned off *V<sub>BUS</sub>*, the operating point when *V<sub>BUS</sub>* is below *vSafe5V* max is referred to as *Stage 1*.
  - b. PD messaging sub-steps:
    - i. The Tester presents *SinkTxOk* immediately after sending Fast Role Swap signal.

- ii. The Tester checks that *FR\_Swap* Message starts to be sent by the UUT within *tFRSwapRx* max plus *tFRSwapInit* max of the start of the Fast Role Swap signal. [TEST.PD.FRS.SNK3.7#1]
- iii. Upon receipt of the *FR\_Swap* Message, the Tester replies with a *GoodCRC* and sends an *Accept* Message to the UUT.
- iv. The Tester changes from presenting Rp to Rd as soon as the operating point has reached Stage 1.
- v. The Tester sends *PS\_RDY* with the Port Power Role field = 0b (Sink).
- vi. The check fails if the *PS\_RDY* is not received within *tPSSourceOn* min. The delay is measured from the time the last bit of the EOP of the *GoodCRC* Message corresponding to the transmitted *PS\_RDY* Message to the time the last bit of the EOP of the *GoodCRC* Message corresponding to the received *PS\_RDY* Message. [TEST.PD.FRS.SNK3.7#2]
- vii. Upon receipt of the *PS\_RDY* Message, the Tester intentionally does not reply with a *GoodCRC* Message.
- viii. The check fails if Error Recovery is not detected within *tCtsErrorRecovery* after the last bit of the EOP of the last retried *PS\_RDY* Message EOP has been received to the time the Error Recovery is detected. [TEST.PD.FRS.SNK3.7#3]

## 9 USB4

### 9.1 USB4 Tests – Data Reset

#### 9.1.1 PD3 Mode

##### TEST.PD.USB4.DRST.1 –Data\_Reset command response of UFP UUT

##### Description:

Tester verifies the UFP UUT provides proper response for the *Data\_Reset* Message from DFP.

##### Test Specific Tester Behavior:

- The Tester is a Source and DFP
- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always rejects *VCONN\_Swap* COMMON.PROC.PD.6

##### Test Conditions:

	Consumer Only	Provider Only, P/C	DRP, C/P
Rev3ChkdSnk	✓		✓

##### Test Procedures:

1. Bring up the UUT with bring-up procedure COMMON.PROC.BU2
2. The Tester sends the *Data\_Reset* Message to UUT
3. The Tester checks for response. [TEST.PD.USB4.DRST.1#1]
  - a. If the Tester receives the *Not\_Supported* Message:
    - i. check that the VIF field Data\_Reset\_Supported = No
    - ii. the test ends here.
  - b. If the Tester receives the *Accept* Message:
    - i. check that the VIF field Data\_Reset\_Supported = Yes,
    - ii. do/simulate Tester Power Cycle  $V_{CONN}$  in accordance with requirements.
  - c. else the test stops here.
4. The Tester sends the *Data\_Reset\_Complete* Message and checks that UFP is not presenting  $V_{CONN}$  after receiving the last bit of *GoodCRC* from UUT for *tCtsInfinite* [TEST.PD.USB4.DRST.1#4]
5. The Tester sends the *Get\_Sink\_Cap* Message and checks that the data role is the same as before the data reset (The Tester is DFP, and UUT is UFP) after receiving the last bit of *GoodCRC* from UUT [TEST.PD.USB4.DRST.1#5]

#### **TEST.PD.USB4.DRST.2 –Data\_Reset command response of UFP UUT, Invalid Sequence**

**Description:** The Tester verifies UFP UUT initiates the error recovery upon failure of Data\_Reset process. The test includes two independent subtests with different ways to invoke Error Recovery

##### **Test Specific Tester Behavior:**

- The Tester is a Source and DFP
- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always rejects *VCONN\_Swap* COMMON.PROC.PD.6

##### **Test Conditions:**

This test is applicable if the VIF parameter “Data\_Reset\_Supported” is set to Yes

	Consumer Only	Provider Only, P/C	DRP, C/P
Rev3ChkdSnk	✓		✓

##### **Test Procedures:**

1. Bring up the UUT with bring-up procedure COMMON.PROC.BU.2
2. The Tester sends the *Data\_Reset* Message to UUT

3. The Tester checks that it receives the *Accept* Message [TEST.PD.USB4.DRST.2#1]. If the message has not been received, the test ends here.
4. Do/simulate Power Cycle  $V_{CONN}$  in accordance with requirements.
5. The Tester sends a *Get\_Sink\_Cap* Message instead of the *Data\_Reset\_Complete* Message and checks that UUT performs Error Recovery within *tCtsErrorRecovery* [TEST.PD.USB4.DRST.2#3]
6. The Tester removes all terminations and simulates a disconnection.
7. Bring-up the UUT with bring-up procedure COMMON.PROC.BU.2
8. The Tester sends the *Data\_Reset* Message to UUT
9. The Tester checks that it receives the *Accept* Message [TEST.PD.USB4.DRST.2#2]. If the message has not been received, the test ends here.
10. Do/simulate Power Cycle  $V_{CONN}$  in accordance with requirements.
11. The Tester deliberately fails to send the *Data\_Reset\_Complete* Message and checks that UUT performs Error Recovery within *tDataResetFailUFP* max [TEST.PD.USB4.DRST.2#4].

### **TEST.PD.USB4.DRST.3 –Data\_Reset command response of UFP UUT Sourcing Vconn**

#### Description:

Verify the  $V_{CONN}$  power cycling of UFP upon accepting the *Data\_Reset* Message from DFP

#### Test Specific Tester Behavior:

- The Tester is a Source and DFP
- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always rejects *VCONN\_Swap* COMMON.PROC.PD.6

#### Test Conditions:

This test is applicable if the VIF parameter “Data\_Reset\_Supported” is set to Yes and “VCONN\_Swap\_To\_On\_Supported= Yes” on Vendor information file

	Consumer Only	Provider Only, P/C	DRP, C/P
Rev3ChkdSnk	✓		✓

#### Test Procedures:

1. Bring up the UUT with bring-up procedure COMMON.PROC.BU.2
2. Tester sends *VCONN\_Swap* message to bring UFP UUT as Vconn provider. Tester checks the response. [TEST.PD.USB4.DRST.3#1]



- a. If the UUT responds with a *Wait* Message and the Tester resends *VCONN\_Swap* every  $tVCONN_{SwapWait} + tCtsWait$  for  $nCtsWaitCount$  times. The check fails and the test ends here if the Tester has resent *VCONN\_Swap* for  $nCtsWaitCount$  times.
  - b. The check fails and test ends here if the UUT does not complete the *VCONN\_Swap*.
3. The Tester sends the *Data\_Reset* Message to UUT
4. The Tester checks for response.
  - a. If the tester doesn't receive *Accept*, the check fails, and the test ends here. [TEST.PD.USB4.DRST.3#2]
  - b. If the tester receives *Accept*, tester perform below steps:
    - I. Check fails if the  $V_{CONN}$  voltage is not below  $vRaReconnect$  in  $tVconnZero$ . This is measured from the last bit of the *GoodCRC* acknowledging the *Accept* Message in response to the *Data\_Reset* Message [TEST.PD.USB4.DRST.3#3]
    - II. Check fails if the first bit of the preamble of the *PS\_RDY* Message has not started within minimum of  $tVconnZero$ . This is measured from the last bit of the *GoodCRC* Message acknowledging the *Accept* Message in response to the *Data\_Reset* Message [TEST.PD.USB4.DRST.3#4]
    - III. Check fails if the first bit of the preamble of the *PS\_RDY* Message is received before  $V_{CONN}$  voltage is below  $vRaReconnect$  [TEST.PD.USB4.DRST.3#7]
5. The Tester sends the *Data\_Reset\_Complete* Message and checks if UFP is not presenting  $V_{CONN}$  at the end of the *GoodCRC* from UUT for  $tCtsInfinite$  [TEST.PD.USB4.DRST.3#6]

#### *TEST.PD.USB4.DRST.4 –Data\_Reset command response of UFP UUT Sourcing Vconn – Invalid Sequence*

##### Description:

Verify that UFP initiates the error recovery upon invalid sequence of UFP Vconn power cycle.

##### Test Specific Tester Behavior:

- The Tester is a Source and DFP
- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always rejects *VCONN\_Swap* COMMON.PROC.PD.6

### Test Conditions:

This test is applicable if the VIF parameter “Data\_Reset\_Supported” is set to Yes and “VCONN\_Swap\_To\_On\_Supported= Yes” on Vendor information file

	Consumer Only	Provider Only, P/C	DRP, C/P
Rev3ChkdSnk	✓		✓

### Test Procedures:

1. Bring up the UUT with bring-up procedure COMMON.PROC.BU.2
2. The Tester sends the *VCONN\_Swap* Message to bring UFP UUT as Vconn provider. The Tester checks the response. [TEST.PD.USB4.DRST.4#1]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *VCONN\_Swap* every *tVCONNSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *VCONN\_Swap* for *nCtsWaitCount* times.
  - b. The check fails and test ends here if the UUT does not complete the *VCONN\_Swap*.
3. The Tester sends the *Data\_Reset* Message to UUT
4. The Tester checks if the *Accept* Message is not received. [TEST.PD.USB4.DRST.4#2]. If the *Accept* Message is not received the test ends here.
5. The Tester sends the *VCONN\_Swap* Message *tCtsErrorRecovery* after the end of *GoodCRC* following *Accept* and checks that the UUT does Error Recovery in *tCtsErrorRecovery* [TEST.PD. USB4.DRST.4#3]

### *TEST.PD.USB4.DRST.5 – Data\_Reset command response of DFP UUT Sourcing Vconn*

### Description:

The Tester verifies DFP UUT accepts the *Data\_Reset* Message and provides the proper response.

Verify the V<sub>CONN</sub> power cycle sequence and DFP remains the V<sub>CONN</sub> source after completion of data\_reset

### Test Specific Tester Behavior:

- The Tester is a Sink and UFP
- Tester emulates active cable.
- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message

- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always rejects *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

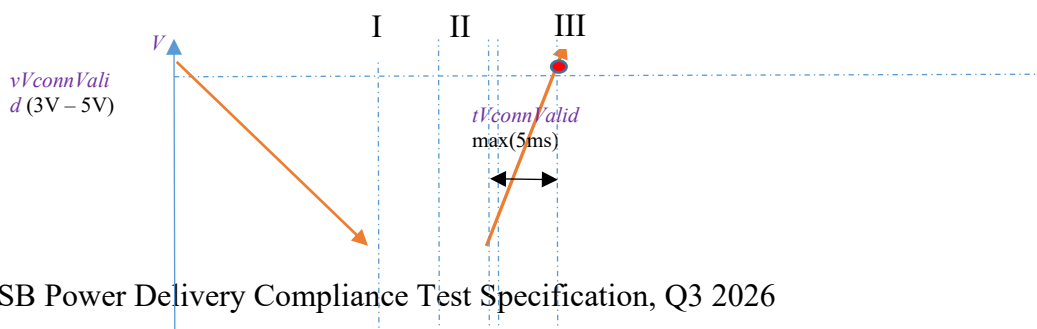
#### Test Conditions:

If the parameter “Data\_Reset\_Supported = YES” on the vendor file, this test is applicable.

	Consumer Only, C/P	Provider Only	DRP, P/C
Rev3ChkdSrc		✓	✓

#### Test Procedures:

1. Bring-up the UUT with bring-up procedure COMMON.PROC.BU.1.
2. The Tester send the *Data\_Reset* Message to UUT. The Tester checks for response.
  - a. If the Tester doesn't receive *Accept*, the check fails. The test ends here. [TEST.PD. USB4.DRST.5#1]
  - b. If the Tester receives *Accept* and VIF field Captive\_Cable is set to NO, the Tester performs below steps.
    - I. The Tester checks that  $V_{CONN}$  is below *vRaReconnect* by *tVconnZero* max. [TEST.PD. USB4.DRST.5#2]. The *tVconnZero* is measured from the last bit of *GoodCRC* acknowledging *Accept*.
    - II. The Tester checks that  $V_{CONN}$  stays below *vRaReconnect* for at least *tVconnReapplied* min and at most *tVconnReapplied* max [TEST.PD. USB4.DRST.5#3]
    - III. The Tester checks the  $V_{CONN}$  voltage reaches valid range by *tVconnValid* max. [TEST.PD. USB4.DRST.5#5]. The time is measured from  $V_{CONN}$  rising above *vRaReconnect* to *vVconnValid* min.



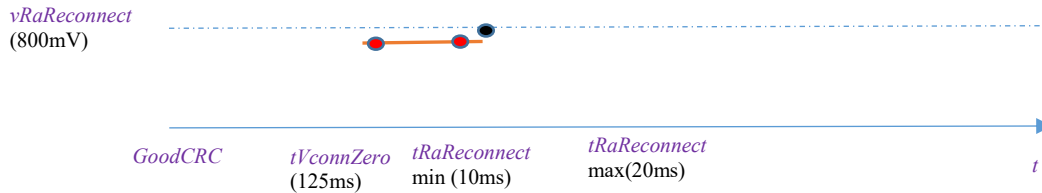


Figure 8  $V_{CONN}$  measurements during Data Reset

**NOTE:** *vRaReconnect* and *vVconnValid* are defined in the USB Type-C specification

3. The Tester checks that it receives *Data\_Reset\_Complete* in *tDataReset* max, else the test ends here. [TEST.PD. USB4.DRST.5#8].
4. The Tester checks that it  $V_{BUS}$  is not changed, and it is in valid range throughout the *Data\_Reset* process. [TEST.PD. USB4.DRST.5#7]
5. If the VIF “USB4\_Supported= Yes”, the Tester checks *Discover Identity* on SOP seen within *tCtsDataResetDiscovEntry* after receiving *Data\_Reset\_Complete* [TEST.PD. USB4.DRST.5#10]

#### TEST.PD. USB4.DRST.6 –Data\_Reset command response of DFP UUT, UFP Sourcing Vconn

##### Description:

The Tester verifies DFP UUT accepts the *Data\_Reset* Message and provides the proper response when UFP sourcing  $V_{CONN}$ .

Verify the  $V_{CONN}$  power cycle sequence and DFP becomes the  $V_{CONN}$  source after completion of *Data\_Reset*

##### Test Specific Tester Behavior:

- The Tester is a Sink and UFP
- The tester emulates active cable.
- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always rejects *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

##### Test Conditions:

This test is applicable if the VIF parameter “Data\_Reset\_Supported” is set to Yes and “VCONN\_Swap\_To\_OFF\_Supported= Yes” on Vendor information file

	Consumer Only, C/P	Provider Only	DRP, P/C
Rev3ChkdSrc		✓	✓

#### Test Procedures:

1. Bring-up the UUT with bring-up procedure COMMON.PROC.BU.1.
2. Tester sends *VCONN\_Swap* message to become Vconn provider. Tester checks the response. [TEST.PD.USB4.DRST.6#1]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *VCONN\_Swap* every *tVCONNswapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *VCONN\_Swap* for *nCtsWaitCount* times.
  - b. The check fails and test ends here if the UUT does not send an *Accept* Message.
  - c. Tester sends *PS\_RDY* upon accept of VconnSwap
3. The Tester sends the *Data\_Reset* Message to UUT. The Tester checks for response.
  - a. If the Tester doesn't receive the *Accept* Message, the check fails, and the test stops here [TEST.PD. USB4.DRST.6#2]
  - b. If the Tester receives *Accept* Message, the tester performs below steps.
    - I. The Tester takes  $V_{CONN}$  to *vRaReconnect* (or simulates), and then sends *PS\_RDY* after *tVconnZero* minus 1ms. The *tVconnZero* is measured from the last bit of *GoodCRC* acknowledging *Accept*.
    - II. The Tester checks that Vconn stays below *vRaReconnect* for at least *tVconnReapplied* min and at most *tVconnReapplied* max. The time is measured from the last bit of *GoodCRC* acknowledging *PS\_RDY* to  $V_{CONN}$  rising above *vRaReconnect*. [TEST.PD. USB4.DRST.6#3]
  - c. If the Tester receives *Accept* Message and the VIF field Captive\_Cable is set to NO, the tester performs below steps.
    - I. The Tester checks that  $V_{CONN}$  rises above *vRaReconnect* by *tVconnReapplied* max. [TEST.PD. USB4.DRST.6#4]
    - II. The Tester checks that  $V_{CONN}$  is reaches valid by *tVconnvalid* max. [TEST.PD. USB4.DRST.6#5]. The time is measured between  $V_{CONN}$  crossing *vRaReconnect* to min of *vVconnValid*.
4. The Tester checks that it receives the *Data\_Reset\_Complete* in *tDataReset* max time, else the test ends here. [TEST.PD. USB4.DRST.6#8].
5. The Tester checks that  $V_{BUS}$  is not changed, and *Data\_Reset\_Complete* in *tDataReset* max, else the test ends here [TEST.PD. USB4.DRST.6#7].
6. If the VIF “USB4\_Supported= Yes”, check *Discover Identity* SOP seen within *tCtsDataResetDiscovEntry* [TEST.PD. USB4.DRST.6#9]

*TEST.PD.USB4.DRST.7 –Data\_reset command response of DFP UUT, UFP Sourcing Vconn- VCONNDischargeTimer expiry check.*

Description:

The Tester verifies DFP UUT accepts the *Data\_Reset* Message and provides the proper response when UFP sourcing vconn. Tester verifies implementation of Vconn Discharge Timer on DFP.

Test Specific Tester Behavior:

- The Tester is a Sink and UFP
- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always rejects *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

Test Conditions:

This test is applicable if the VIF parameter “Data\_Reset\_Supported” is set to Yes and “VCONN\_Swap\_To\_Off\_Supported= Yes” on Vendor information file

	Consumer Only, C/P	Provider Only	DRP, P/C
Rev3ChkdSrc		✓	✓

Test Procedures:

1. Bring-up the UUT with bring-up procedure COMMON.PROC.BU.1.
2. Tester sends *VCONN\_Swap* message to become Vconn provider. Tester checks the response. [TEST.PD.USB4.DRST.7#1]
  - a. If the UUT responds with a *Wait* Message and the Tester resends *VCONN\_Swap* every  $tVCONNSwapWait + tCtsWait$  for  $nCtsWaitCount$  times. The check fails and the test ends here if the Tester has resent *VCONN\_Swap* for  $nCtsWaitCount$  times.
  - b. The check fails and test ends here if the UUT does not send an *Accept* Message.
  - c. Tester sends *PS\_RDY* upon accept of VconnSwap
3. The Tester sends the *Data\_Reset* Message to UUT. The Tester checks for response.
  - a. If the Tester doesn't receive *Accept*, the check fails, and the test ends here [TEST.PD.USB4.DRST.7#2]
  - b. If the tester receives *Accept* Message, the tester performs below steps.

- I. The Tester intentionally do not send *PS\_RDY* after dropping (or simulating)  $V_{CONN}$ .
- II. The Tester checks for UUT going Error Recovery in within *tVCONNSourceDischarge* max. [TEST.PD.USB4.DRST.7#3] The *tVCONNSourceDischarge* is measured from the last bit of the *GoodCRC* acknowledging the *Accept* Message in response to the Data\_Reset Message.

## 9.2 USB4 Tests – Enter USB4

### 9.2.1 PD3 Mode

#### TEST.PD.USB4.EUSB.1 – Enter\_USB Message response of UFP UUT-Valid Mode

##### Description:

The Tester as USB4 DFP verifies that the USB4 capable UFP accepts *Enter\_USB* Message and provides the proper response.

##### Test Specific Tester Behavior:

Unless a Test Specific Tester Behavior is specified, this section uses the following behavior:

- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message.
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

##### Test Conditions:

If parameter “USB4\_UFP\_Supported” is set to Yes in the vendor file, this test is applicable.

	Consumer Only, C/P	Provider Only, P/C	DRP
Rev3ChkdSnk	✓		✓
Rev3ChkdSrc		✓	

##### Test Procedures:

1. The tester starts with two possible bring-up procedures.
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only / DRP or C/P, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 and does the below steps to bring the UUT as UFP.
    - a. Tester sends a *DR\_Swap* Message and check response [TEST.PD.USB4.EUSB.1#1]



- i. If the UUT responds with a *Wait* Message and the Tester resends *DR\_Swap* every *tDRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *DR\_Swap* for *nCtsWaitCount* times.
  - ii. If the UUT not responded with *Accept*, the check fails and test ends here.
2. The Tester sends a *Discover Identity* Command to the UUT, using SOP. The Tester checks the response. [TEST.PD.USB4.EUSB.1#3]
  - a. The check fails if the UUT responds with a “Responder NAK”.
  - b. The check fails if the UUT responds with *Not\_Supported*
  - c. If the UUT interrupts the Discovery Process (by sending anything but a *Discover Identity* Response message), the Tester resends *Discover Identity* Command *tVDMBusy* min after it is in PE\_SNK\_RDY for *nBusyCount* times. The check fails if the Tester has resent *Discover Identity* Command for *nBusyCount* times.
  - d. The check fails if the UUT responds with a “Responder BUSY” message and the Tester has resent *Discover Identity* Command for *nBusyCount* times every *tVDMBusy* min.
  - e. The check fails if the UUT does not send “Responder ACK”
3. The Tester formulates the EUDO the EUDO as described below. The Tester sends an *Enter\_USB* Message to the UUT using SOP with host present bit = 0 to make speculative connection.

USB Mode: USB4, Cable speed: USB4 Gen4, Cable type: Passive, Cable current: 3A. Host Present = 0, Set everything else to 0.

4. The Tester checks the response from UFP. Testers check the below:
  - a. The check fails if the tester receives any message other than *Accept*, *Reject* or *Wait* Message, and the test stops here. If the *Reject* Message is received, the Tester reports a warning, and the test stops here. [TEST.PD.USB4.EUSB.1#4]
  - b. If the UUT responds with the *Wait* Message, the Tester resends Request Message every *tSinkRequest* for *nCtsWaitCount* times or until the *Accept* Message is received. The check fails if the UUT responds with Wait Message after *nCtsWaitCount* retries, and the test stops here. [TEST.PD.USB4.EUSB.1#6].
  - c. The tester issues a warning if UUT initiated any AMS in *tEnterUSB*. For Consumer Only/C/P and the DRP the *tEnterUSB* is measured from last bit of *GoodCRC* acknowledging source cap from tester. For Provider only, P/C, it is measured from *GoodCRC* for the *Accept* response of DR\_SWAP

5. The Tester sends an *Enter\_USB* Message to the UUT using SOP with EUDO set with below parameters to make USB4PDhost connection.

USB Mode: USB4, Cable speed: USB4 Gen4, Cable type: Passive, Cable current: 3A. Host Present = 1, Set everything else to 0.

6. The Tester checks the response from UFP. The check fails if the tester receives any message other than *Accept*, *Reject* or *Wait* Message. [TEST.PD.USB4.EUSB.1#5].

## TEST.PD.USB4.EUSB.2 – Enter\_USB Message response of UFP UUT-Invalid Mode

### Description:

The Tester as USB4 DFP verifies that the USB4 capable UFP Rejects invalid *Enter\_USB* request and provides the proper response.

### Test Specific Tester Behavior:

Unless a Test Specific Tester Behavior is specified, this section uses the following behavior:

- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- The Tester always provides a NAK response for *Discover Identity* COMMON.PROC.PD.7
- The Tester always provides a NAK response for *Discover SVIDs* COMMON.PROC.PD.8

### Test Conditions:

If parameter “USB4\_UFP\_Supported” is set to Yes in the vendor file, this test is applicable.

	Consumer Only, C/P	Provider Only, P/C	DRP
Rev3ChkdSnk	✓		✓
Rev3ChkdSrc		✓	

### Test Procedures:

1. The tester starts with two possible bring-up procedures.
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only / DRP or C/P, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 and does the below steps to bring the UUT as UFP.
    - I. Tester sends a *DR\_Swap* Message and check response [TEST.PD.USB4.EUSB.2#1]
      - i. If the UUT responds with a *Wait* Message and the Tester resends *DR\_Swap* every *tDRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *DR\_Swap* for *nCtsWaitCount* times.
      - ii. If the UUT not responded with *Accept*, the check fails and test end here.

2. The Tester sends a *Discover Identity* Command to the UUT, using SOP. The Tester checks the response. [TEST.PD.USB4.EUSB.2#3]

- f. The check fails if the UUT responds with a “Responder NAK”.
- g. The check fails if the UUT responds with *Not Supported*
- h. If the UUT interrupts the Discovery Process (by sending anything but a *Discover Identity* Response message), the Tester resends *Discover Identity* Command *tVDMBusy* min after it is in PE\_SNK\_RDY for *nBusyCount* times. The check fails if the Tester has resent *Discover Identity* Command for *nBusyCount* times.
- i. The check fails if the UUT responds with a “Responder BUSY” message and the Tester has resent *Discover Identity* Command for *nBusyCount* times every *tVDMBusy* min.
- j. The check fails if the UUT does not send “Responder ACK”

3. The Tester sends an *Enter\_USB* Message to the UUT using SOP. The Tester formulates the EUDO with invalid mode as described below:

USB Mode: Reserved (111b), Cable speed: USB2, Cable type: Passive, Cable current: 3A. Host Present = 0, Set everything else to 0.

4. The Tester checks the response from UFP. Testers checks the below

- a. The check fails if the tester receives any message other than *Reject*. [TEST.PD.USB4.EUSB.2#4].
- b. The tester issues a warning if UUT initiated any AMS in *tEnterUSB*. For Consumer Only/C/P and the DRP the *tEnterUSB* is measured from last bit of *GoodCRC* acknowledging source cap from tester. For Provider only, P/C, It is measured from EOP of *GoodCRC* for the *Accept* response of DR\_SWAP

#### *TEST.PD.USB4.EUSB.3 – Enter\_USB Flow-USB4 DFP Connected to USB4 UFP using an Active Cable*

Description: The Tester verifies the *Enter\_USB* flow of USB4 Capable DFP UUT when it is connected to a USB4 capable UFP using a USB4 Capable Active cable.

#### Test Specific Tester Behavior:

- The Tester emulates USB4 capable UFP, PD Peripheral device.
- The Tester emulates USB4 Gen4 Active cable. Highest Speed USB4 Gen4, Current – 3A, V<sub>BUS</sub> Max-20V, Cable Latency -0001b, SOP” Controller Present=1.
- The Tester always accepts DR\_SWAP\_to\_UFP and rejects DR\_Swap\_to\_DFP
- The tester always accepts the Vconn\_Swap
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always provides ACK response with below settings for *Discover Identity* Command as in COMMON.PROC.PD.7.
  - On the SOP’

- ID header VDO-> Product Type UFP: Active Cable, Connector Type: USB TypeC plug, Modal operation supported = No, USB VendorID= 0x1A0A. USB ProductID=0x0301. All other values are 0.
  - Active Cable VDO 1-> Connector: USB TypeC, Max Current 3A, Max Voltage=20V, Max Speed= USB4 Gen4. Cable termination = 11b. Cable latency=0001b, SOP” Controller Present=1, V<sub>BUS</sub> Through Cable = 1b, remaining fields are 0
  - Active Cable VDO 2-> Max Temp=70, Shutdown Temp=80, USB Gen=1, remaining all fields are 0
- On the SOP
    - ID header VDO->Product Type UFP: PDUSB Peripheral, Connector Type: USB TypeC Receptacle
    - UFP VDO->Device capability: USB4 Device capable, Highest Speed: USB4 Gen4
- The Tester always provides NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### Test Conditions:

If USB4\_DFP\_Supported is set to Yes in the vendor file, this test is applicable.

	Consumer Only, C/P	Provider Only	DRP, P/C
Rev3ChkdSrc		✓	✓
Rev3ChkdSnk	✓		

#### Test Procedures:

##### 1. There are two possible bring-up procedures

- a. For Provider Only, DRP, P/C, bring-up the UUT with bring-up procedure COMMON.PROC.BU.1.
- b. For Consumer Only, C/P bring-up the UUT with be UUT with COMMON.PROC.BU.2 and perform below steps.
  - i. The Tester sends the *DR\_Swap* Message to bring the UUT as DFP. The Tester checks the response. [TEST.PD.USB4.EUSB.3#1]
    - a. If the UUT responds with a *Wait* Message and the Tester resends *DR\_Swap* every *tDRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the Test ends here if the Tester has resent *DR\_Swap* for *nCtsWaitCount* times.
    - b. The check fails if the UUT not accepted the *DR\_Swap* Message, the Test ends here.

##### 2. If the Tester receives the *Enter\_USB* Message on SOP’, the Tester performs the below checks.

- a. Verify the USB mode on EUDO is set to USB4[TEST.PD.USB4.EUSB.3#2]
  - b. Verifies the reserved fields on EUDO(b31, b27,b24,b12-0) are set to 0. [TEST.PD.USB4.EUSB.3#3]
  - c. The Tester sends the *Accept* response on SOP’.
3. If the Tester receives the *Enter\_USB* Message on SOP’’, the Tester performs the below checks.
  - a. Verify the USB mode on EUDO is set to USB4[TEST.PD.USB4.EUSB.3#4]
  - b. Verifies the reserved fields on EUDO(b31, b27,b24,b12-0) are set to 0[TEST.PD.USB4.EUSB.3#5]
  - c. The Tester sends the *Accept* response on SOP’’.
4. If the Tester receives the *Enter\_USB* Message on SOP, the Tester performs the below checks.
  - a. The *Enter\_USB* Message is received with in *tEnterUSB* [TEST.PD. USB4.EUSB.3#6]  
For Provider, DRP, P/C UUTs, the *tEnterUSB* is measured from last bit of *GoodCRC* acknowledging source cap from the UUT. For C, C/P UUTs, the *tEnterUSB* is measured from last bit of *GoodCRC* acknowledging the *Accept* Message in response for *DR\_Swap*
  - b. The Tester verifies below parameters of EUDO matching to the tester’s UFP VDO and Cable VDOs. The Tester sends the *Accept* response if the checks are true. The Tester sends the *Reject* response if any of the below checks are false. [TEST.PD.USB4.EUSB.3#7]
    - a. USB Mode = USB4
    - b. Cable Speed = USB4 Gen3 or Gen4,
    - c. Cable Type =Active Re-driver,
    - d. Cable Current=3A
    - e. Reserved fields are set to 0 (b31, b27, b24, b12-0)
  - c. The Tester checks the consistency of EUDO fields against the VIF. The check fails if any of the fields are not matching. [TEST.PD.USB4.EUSB.3#9]
    - i. USB4 DRD should set to 1 if USB4\_UFP\_Supported =YES. USB4 DRD should set to 0 if USB4\_UFP\_Supported =No.
    - ii. USB3 DRD should set to 1, if Type\_C\_Can\_Act\_As\_Device is set to 1 and Device Speed is USB3 Gen 1 or above,
    - iii. PCIe Support field setting matches to the VIF field  
USB4\_PCIe\_Tunneling\_Supported
    - iv. DP Support field setting matches to the VIF field  
USB4\_DP\_Tunneling\_Supported
    - v. TBT Support field setting matches to the VIF field  
USB4\_TBT3\_Compatibility\_Supported = Yes
    - vi. If Product\_Type\_DFP\_SOP=PDUSB Host, verify that the Host Present bit set to 1
5. The Tester waits for *tUSB4Timeout*. The test fails if the tester doesn’t receive the *Enter\_USB* on SOP. The *tUSB4Timeout* is measured from sink(tester) attach for Provider, DRP, P/C. For C,

C/P it is measured from last bit of ‘*GoodCRC*’ acknowledging *Accept* response for the ‘DR\_Swap’ message. [TEST.PD.USB4.EUSB.3#8]

#### *TEST.PD.USB4.EUSB.4 – DR\_Swap after Entering USB4 Mode entry.*

##### Description:

Verify the USB4 capable DRD accepting DR\_SWAP after entering USB4 mode. Verify the UUT performs the USB4 device discovery and mode entry to USB4 after DR\_SWAP

##### Test Specific Tester Behavior:

Unless a Test Specific Tester Behavior is specified, this section uses the following behavior:

- Tester emulates as USB4 Host and USB4 Peripheral with DRD Capable
- The Tester always replies with a *Reject* Message to a *DR\_Swap* Message
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always accepts *VCONN\_Swap* COMMON.PROC.PD.6
- When acting as DFP, Tester emulates USB4 Host and send *Enter\_USB* with below setting EUDO.
  - USB Mode: USB4, Cable speed: USB4 Gen4, Cable type: Active Re-timer, Cable current: 3A. SOP” Controller Present = 1, USB4 DRD = 1, Host Present = 1, Set everything else to 0.
- The Tester always provides ACK response with the settings below for *Discover Identity* Command as in COMMON.PROC.PD.7.
  - On the SOP’
    - ID header VDO-> Product Type UFP: Active Cable, Connector Type: USB TypeC plug, Modal operation supported = No
    - Active Cable VDO 1:
      - VDO Version (B23...B21) 1.3: 011b,
      - Connector: USB Type-C(B19...18) USB Type-C: 10b,
      - Cable Latency (B16...13)<10ns: 0001b,
      - Maximum V<sub>BUS</sub> Voltage (B10...9) 20V: 00b,
      - V<sub>BUS</sub> Current Handling (B6...5) Max Current 3A: 01b
      - USB Highest Speed (B2...0) USB4 Gen4: 100b,
      - All Remaining fields are 0
    - Active Cable VDO 2:
      - Max Temp (B31...24) 70C: 0100 0110b,
      - Shutdown Temp(B23...16) 80C, 0101 0000b,
      - USB Supported (B8)Supported: 0b,
      - USB Gen (B0) Gen2: 1b,
      - All remaining fields are 0

- On the SOP
  - ID header VDO->Product Type UFP: PDUSB Peripheral, Connector Type: USB TypeC Receptacle
  - UFP VDO->Device capability: USB4 Device capable, , Highest Speed: USB4 Gen4
- The Tester always provides NAK response for *Discover SVIDs* COMMON.PROC.PD.8

#### Test Conditions:

If parameter “USB4\_UFP\_Supported” and USB4\_DFP\_Supported” is set to Yes in the vendor file, this test is applicable.

	Consumer Only, C/P	Provider Only, P/C	DRP
Rev3ChkdSnk	✓		✓
Rev3ChkdSrc		✓	

#### Test Procedures:

1. The tester starts with two possible bring-up procedures.
  - a. The UUT has VIF field PD\_Port\_Type set to Consumer Only / DRP or C/P, the Tester runs bring-up procedure with the UUT as a Sink COMMON.PROC.BU.2
  - b. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1 and does the below steps to bring the UUT as UFP.
    - a. Tester sends a *DR\_Swap* Message and check response [TEST.PD. USB4.EUSB.4#1]
      - i. If the UUT responds with a *Wait* Message and the Tester resends *DR\_Swap* every *tDRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the test ends here if the Tester has resent *DR\_Swap* for *nCtsWaitCount* times.
      - ii. If the UUT did not respond with *Accept*, the check fails and test end here.
2. The Tester sends an *Enter\_USB* Message to the UUT using SOP and check the response from UUT. The check fails if the tester receives any message other than *Accept*, *Reject* or *Wait*, and test stops here. If *Reject* Message is received, the Tester reports a warning, and the test stops here [TEST.PD. USB4.EUSB.4#3]
3. if the UUT responds with *Wait* Message, the Tester resends *Request* message every *tSinkRequest* for *nCtsWaitCount* times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after *nCtsWaitCount* retries [TEST.PD. USB4.EUSB.4#11].
4. The Tester sends *DR\_Swap* Message. The Tester checks the response.



5. The check fails if the Tester does not receive Hard Reset signal within *tProtErrHardReset* of the last EOP of the *GoodCRC* sent in response to the *DR\_Swap* Message. [TEST.PD.USB4.EUSB.4#4].

#### *TEST.PD.USB4.EUSB.5 – tEnterUSBWait check for USB4 DFP*

**Description:** The UUT as USB4 DFP Shall wait *tEnterUSBWait* after receiving the *Wait* Message to *Enter\_USB* Message before sending another the *Enter\_USB* Message.

#### Test Specific Tester Behavior:

The Tester emulates USB4 capable UFP, PD Peripheral device.

- The Tester emulates USB4 Gen4 passive cable. Highest Speed USB4 Gen4, Current – 3A, V<sub>BUS</sub> Max-20V, Cable Latency -0001b, SOP' Controller Present=1.
- The Tester always accepts DR\_SWAP\_to\_UFP and rejects DR\_Swap\_to\_DFP
- The tester always accepts the *VCONN\_Swap*
- The Tester always rejects *PR\_Swap* COMMON.PROC.PD.5
- The Tester always provides ACK response with below settings for *Discover Identity* Command as in COMMON.PROC.PD.7.
  - o on the SOP'
    - ID header VDO
      - Product Type UFP: Passive Cable,
      - Connector Type: USB TypeC plug,
      - Modal operation supported = No,
      - USB VendorID = 0x1A0A,
      - All other values are 0
    - Passive Cable VDO
      - Connector: USB TypeC,
      - Max Current = 3A,
      - Max Voltage =20V,
      - Max Speed = USB4 Gen4,
      - Cable termination = 00b,
      - Cable latency=0001b,
      - remaining fields are 0.
  - o on the SOP
    - ID header VDO
      - Product Type UFP: PDUSB Peripheral,
      - Connector Type: USB TypeC Receptacle.
    - UFP VDO
      - Device capability: USB4 Device capable,
      - Highest Speed: USB4 Gen4.
- The Tester always provides NAK response for *Discover SVIDs* COMMON.PROC.PD.8

### Test Conditions:

If USB4\_DFP\_Supported is set to Yes in the vendor file, this test is applicable

	Consumer Only, C/P	Provider Only, P/C	DRP
Rev3ChkdSnk		✓	
Rev3ChkdSrc	✓		✓

### Test Procedures:

1. There are two possible bring-up procedures

- For Provider Only, DRP, P/C, bring-up the UUT with bring-up procedure COMMON.PROC.BU.1.
- For Consumer Only, C/P bring-up the UUT with be UUT with COMMON.PROC.BU.2 and perform below steps:

The Tester sends *DR\_Swap* Message to bring the UUT as DFP. The Tester checks the response. [TEST.PD.USB4.EUSB.5#1]

- If the UUT responds with a *Wait* Message and the Tester resends *DR\_Swap* every *tDRSwapWait* + *tCtsWait* for *nCtsWaitCount* times. The check fails and the Test ends here if the Tester has resent *DR\_Swap* for *nCtsWaitCount* times.
- The check fails if the UUT not accepted the *DR\_Swap*, the Test ends here.

2. If the Tester receives the *Enter\_USB* Message on SOP', the Tester performs the below checks:

- Verify the USB mode on EUDO is set to USB4 (010b) [TEST.PD.USB4.EUSB.5#2]
- Verifies the reserved fields on EUDO (b31, b27,b24,b12-0) are set to 0. [TEST.PD.USB4.EUSB.5#3]
- The Tester sends the *Accept* response on SOP'.

3. The Tester waits for *tUSB4Timeout*. The test fails if the tester doesn't receive the *Enter\_USB* on SOP. The *tUSB4Timeout* is measured from sink(tester) attach for Provider, DRP, P/C. For C, C/P it is measured from last bit of *GoodCRC* acknowledging *Accept* response for the *DR\_Swap* message. [TEST.PD.USB4.EUSB.5#4]

4. If the Tester receives the *Enter\_USB* Message on SOP, the Tester responds with *Wait* Message.

5. The Tester checks, UUT does not send another *Enter\_USB* message again within *tEnterUSBWait* min

time after sending the *EOP* of a *Wait* Message in response to first *Enter\_USB* Message. The check fails if the UUT sends another *Enter\_USB* message within *tEnterUSBWait* time.

[TEST.PD.USB4.EUSB.5#5].

## 9.3 USB4 Tests – Cable Response

### 9.3.1 PD3 Mode

#### TEST.PD.USB4.CBL.1 – Enter\_USB Message response of cable UUT-Valid Mode

##### Description:

The Tester as USB4 DFP verifies the *Enter\_USB* Message response from the cable.

##### Test Conditions:

	Consumer Only	Provider Only, DRP, C/P, P/C	VPD	Free Cable	Captive Cable, Si-Only eMarker
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

##### Test Procedures:

1. There are 3 possible bring-up procedures
  - a. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - b. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.
  - c. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.
2. The Tester sends a *Discover Identity* Command to the UUT, using SOP'. The tester checks the response to know the cable information. If the UUT responds with "Responder BUSY" message, the Tester resends *Discover Identity* Command for *nBusyCount* times every *tVDMBusy* min.
3. The Tester formulates the EUDO with USB Mode = USB4 and other fields set to 0. Tester sends *Enter\_USB* on SOP' and check the response.
  - a. If the VIF Field 'Enter\_USB\_Supported = No', the check fails if the tester receives any response message. [TEST.PD.USB4.CBL.1#1]

- b. If the VIF Fields ‘Enter\_USB\_Supported = Yes’ and ‘USB4\_Supported = No’, the check fails if the tester receives any message other than *Reject*. [TEST.PD. USB4.CBL.1#2]
  - c. If the VIF Fields ‘Enter\_USB\_Supported = Yes’ and ‘USB4\_Supported = Yes’, the check fails if the tester receives any message other than *Accept*. [TEST.PD. USB4.CBL.1#3]
4. If the tester received an Active Cable VDO in response to *Discover Identity* request and ‘SOP” Controller Present’ bit is set to 1 on Active cable VDO, the tester sends *Enter\_USB* Message on SOP” with USB Mode on EUDO set to USB4 and other fields set to 0. Tester checks the response from UUT.
- a. If the VIF Field ‘Enter\_USB\_Supported = No’, the check fails if the tester receives any response message. [TEST.PD. USB4.CBL.1#4]
  - b. If the VIF Fields ‘Enter\_USB\_Supported = Yes’ and ‘USB4\_Supported = No’, the check fails if the tester receives any message other than *Reject* [TEST.PD. USB4.CBL.1#5]
  - c. If the VIF Fields ‘Enter\_USB\_Supported = Yes’ and ‘USB4\_Supported = Yes’, the check fails if the tester receives any message other than *Accept* [TEST.PD. USB4.CBL.1#6]

#### TEST.PD. USB4.CBL.2 – Enter\_USB Message response of Cable UUT-Invalid Mode

##### Description:

The Tester as USB4 DFP verifies the *Enter\_USB* Message response of cables for an invalid request.

##### Test Conditions:

	Consumer Only	Provider Only	DRP, C/P, P/C	Free Cable	Captive Cable, Si-Only eMarker
<i>Rev3VcMinEnd1</i>				✓	✓
<i>Rev3VcMaxEnd1</i>				✓	✓
<i>Rev3VcMinEnd2</i>				✓	
<i>Rev3VcMaxEnd2</i>				✓	

##### Test Procedures:

1. There are 3 possible bring-up procedures
  - a. The UUT has VIF field VIF\_Product\_Type set to Cable. The Tester runs bring-up procedure with the UUT as a Cable Plug COMMON.PROC.BU.3
  - b. The UUT has VIF field PD\_Port\_Type set to DRP, Consumer/Provider or Consumer only, and VIF parameter Captive\_Cable is set to YES, and VIF

parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Sink UUT for Captive Cable Test COMMON.PROC.BU.6.

- c. The UUT has VIF field PD\_Port\_Type set to Provider/Consumer or Provider only, and VIF parameter Captive\_Cable is set to YES, and VIF parameter eMarked Captive Cable is set to YES\*. The Tester runs bring-up procedure with the UUT as a Source UUT for Captive Cable Test COMMON.PROC.BU.11.

2. The Tester sends a *Discover Identity* Command to the UUT, using SOP'. The tester checks the response to know the cable information. If the UUT responds with "Responder BUSY" message, the Tester resends *Discover Identity* Command for *nBusyCount* times every *tVDMBusy* min.

3. The Tester formulates the EUDO with USB Mode = "Reserved" and other fields set to 0. Tester sends *Enter\_USB* on SOP' and check the response.

- a. If the VIF Field 'Enter\_USB\_Supported = No', the check fails if the tester receives any response message. [TEST.PD. USB4.CBL.2#1]
- b. If the VIF Fields 'Enter\_USB\_Supported = Yes', the check fails if the tester receives any message other than *Reject* [TEST.PD. USB4.CBL.2#2]

4. If the tester received Active Cable VDO in response to *Discover Identity* request and 'SOP' Controller Present' bit is set to 1 on Active cable VDO, the tester sends *Enter\_USB* Message on SOP" with USB Mode on EUDO set to "Reserved" and other fields set to 0. Tester checks the response from UUT.

- a. If the VIF Field 'Enter\_USB\_Supported = No', the check fails if the tester receives any response message. [TEST.PD. USB4.CBL.2#3]
- b. If the VIF Fields 'Enter\_USB\_Supported = Yes', the check fails if the tester receives any message other than *Reject* [TEST.PD. USB4.CBL.2#4]

## Appendix A: Requirements for Compliance Test Results

The Compliance Test Results **Shall** be output in XML format, compliant with the official USB-IF documentation and schema

Documentation:

Schema: <https://compliance.usb.org/xsd/USB/DataCollection.xsd>

- In the *testRun* element:
  - The following values **Shall** be set within the *spec* element:
    - title: "USB Power Delivery Compliance Test Specification"
    - version: "1.0"
    - revision: "1.2"
  - Inside *tests*, a *test* element **Shall** be created for each Common Check, Common Procedure, and Test
  - If any tests fail, the *score* element **Shall** be set to "fail". Otherwise, it **Shall** be set to "pass".

- In each *test* element:
  - The *tcID* attribute **Shall** match the test number (e.g. “TEST.PD.PROT.SRC.1”)
  - Inside *conditions*, a *condition* element **Shall** be created for each Test Condition identifier
  - If any conditions fail, the *score* element **Shall** be set to “fail”. Otherwise, it **Shall** be set to “pass”. For Common Checks and Common procedures, the absence of any failing instances Shall result in a pass regardless of whether the check was ever performed.
- In each *condition* element:
  - The *conditionID* attribute **Shall** match the test condition (e.g. “Rev2Src”)
  - An element called *checks* **Shall** be created. Within *checks*, a *check* element **Shall** be created for every check identifier. Checks from within any Bring-up Procedure that is references during the test Shall be included.
  - If any checks fail, the *score* element **Shall** be set to “fail”. Otherwise, it **Shall** be set to “pass”.
    - If any check element score results in warning, still the *condition score* element **Shall** be set to “pass”
- In each *check* element:
  - A *checkID* attribute **Shall** match the check identifier (e.g. “TEST.PD.PROT.SRC.1#1”)
  - If the check fails, the *score* element **Shall** be set to “fail”. Otherwise, it **Shall** be set to “pass” or “warning”.
 

**NOTE:** “warning” is a variant of “pass” informing the device vendor that the device is compliant with the current specification, however its behavior is marginal and may need to change for better interoperability with other devices.
  - A *comment* element **Shall** be used to provide detail on the nature of each failure. The format in which this is done may be determined by the test vendor.
    - In Common Checks and Procedures, the *comment* element **Shall** indicate both the number of failures and total number of times the check was performed. The format in which this is done may be determined by the test vendor.

## Appendix B: XML Output Example

```
<?xml version="1.0"?>
<testReport>
  <product>
    <deviceID>0</deviceID>
    <nameVendor>Product Vendor</nameVendor>
    <nameProduct>Product Name</nameProduct>
    <nameModel>0</nameModel>
  </productInfo>
```

```

<vif>
  <![CDATA[
    <?xml version="1.0" encoding="utf-8"?>
    <VIF xmlns="http://usb.org/VendorInfoFile.xsd">
      <VIF_Specification>3.01</VIF_Specification>
      <VIF_App>
        <Vendor>USB-IF</Vendor>
        <Name>Vendor Info File Editor</Name>
        <Version>3.0.0.1</Version>
      </VIF_App>
      <Vendor_Name>Product Vendor</Vendor_Name>
      <Model_Part_Number>Product Name</Model_Part_Number>
      <Product_Revision>0</Product_Revision>
      <TID>0</TID>
      <VIF_Product_Type value="1">Cable</VIF_Product_Type>
      <Certification_Type value="2">Silicon</Certification_Type>
      <Product/>
      <Component>
        <USB4_Supported value="false"/>
        <USB_PD_Support value="true"/>
        <PD_Specification_Revision value="1">Revision
2.0</PD_Specification_Revision>
        <SOP_Capable value="false"/>
        <SOP_P_Capable value="true"/>
        <SOP_PP_Capable value="false"/>
        <SOP_P_Debug_Capable value="false"/>
        <SOP_PP_Debug_Capable value="false"/>
        <XID value="0"/>
        <Data_Capable_As_USB_Host value="false"/>
        <Data_Capable_As_USB_Device value="false"/>
        <Product_Type value="3">passive Cable</Product_Type>
        <Modal_Operation_Supported value="false"/>
        <USB_VID value="0">0000</USB_VID>
        <PID value="0">0000</PID>
        <bcdDevice value="0">0000</bcdDevice>
        <Cable_HW_Vers value="0">0</Cable_HW_Vers>
        <Cable_FW_Vers value="0">0</Cable_FW_Vers>
        <Type_C_To_Type_A_B_C value="2">Type-C®</Type_C_To_Type_A_B_C>
        <Type_C_To_Plug_Receptacle
value="0">Plug</Type_C_To_Plug_Receptacle>
        <Cable_Latency value="1">&lt;10ns</Cable_Latency>
        <Cable_Termination_Type value="0">Both ends passive, VCONN not
required</Cable_Termination_Type>
        <VBUS_Through_Cable value="true"/>
        <Cable_VBUS_Current value="2">5A</Cable_VBUS_Current>
        <Cable_SOP_PP_Controller value="false"/>
        <Cable_Superspeed_Support value="2">USB 3.1 Gen
2</Cable_Superspeed_Support>
      </Component>
    </VIF>
  ]>
</vif>
</productInfo>
</product>
<components>
  <component>
    <technology/>

```



```

<subject/>
<value>1</value>
<testTool>
  <deviceID/>
  <nameVendor>Test Vendor</nameVendor>
  <nameProduct>Test Product</nameProduct>
  <nameModel>XYZ</nameModel>
  <versionHW>1.23</versionHW>
  <versionSW>1.23 b456</versionSW>
  <versionFW>1.23</versionFW>
  <serialNumber>12345</serialNumber>
  <dateCalibration/>
  <configuration/>
  <description/>
  <comments/>
  <testRuns>
    <testRun timestamp="2020-08-30T12:34:56">
      <spec>
        <title>USB Power Delivery Compliance Test Specification</title>
        <revision>q2 2023</revision>
        <version>1.0</version>
      </spec>
      <score value="fail"/>
      <test tcID="TEST.PD.PHY.ALL.5">
        <title>Unrecognized Message</title>
        <score value="pass"/>
        <conditions>
          <condition conditionID="Rev2VcMinEnd1">
            <score value="pass"/>
            <checks>
              <check checkID="TEST.PD.PHY.ALL.5#1">
                <score value="pass"/>
                <comment/>
              </check>
              <check checkID="TEST.PD.PHY.ALL.5#2">
                <score value="n/a"/>
                <comment/>
              </check>
            </checks>
          </condition>
          <condition conditionID="Rev2VcMaxEnd1">
            <score value="pass"/>
            <checks>
              <check checkID="TEST.PD.PHY.ALL.5#1">
                <score value="pass"/>
                <comment/>
              </check>
              <check checkID="TEST.PD.PHY.ALL.5#2">
                <score value="n/a"/>
                <comment/>
              </check>
            </checks>
          </condition>
          <condition conditionID="Rev3VcMinEnd1">
            <score value="pass"/>
            <checks>
              <check checkID="TEST.PD.PHY.ALL.5#1">

```

```

        <score value="pass"/>
        <comment/>
    </check>
    <check checkID="TEST.PD.PHY.ALL.5#2">
        <score value="n/a"/>
        <comment/>
    </check>
</checks>
</condition>
<condition conditionID="Rev3VcMaxEnd1">
    <score value="pass"/>
    <checks>
        <check checkID="TEST.PD.PHY.ALL.5#1">
            <score value="pass"/>
            <comment/>
        </check>
        <check checkID="TEST.PD.PHY.ALL.5#2">
            <score value="n/a"/>
            <comment/>
        </check>
    </checks>
</condition>
</conditions>
</test>
<test tcID="COMMON.CHECK.PD.3">
    <title>Check Request Message</title>
    <score value="fail"/>
    <conditions>
        <condition conditionID="">
            <score value="fail"/>
            <checks>
                <check checkID="TEST.PD.PHY.ALL.5#1">
                    <score value="fail"/>
                    <comment>Failed in 1 of 3 instances. Failures occurred
during: TEST.PD.PHY.ALL.5</comment>
                </check>
                <check checkID="TEST.PD.PHY.ALL.5#1">
                    <score value="pass"/>
                    <comment/>
                </check>
            </checks>
        </condition>
    </conditions>
</test>
</testRun>
</testRuns>
</testTool>
</component>
</components>
</testReport>

```

## Appendix C: Power Disconnection for FRS Initial Source Testing

The initial Source testing requires the ability to trigger the Source to send a Fast Role Swap signal. Two methods of performing this are required:

1. Interrupting the AC mains / wall power to the UUT.
2. Interrupting the DC power between the UUT and its power supply. This method is important because it is more likely to cause issues during testing, as there is less possibility of early warning of the interruption.

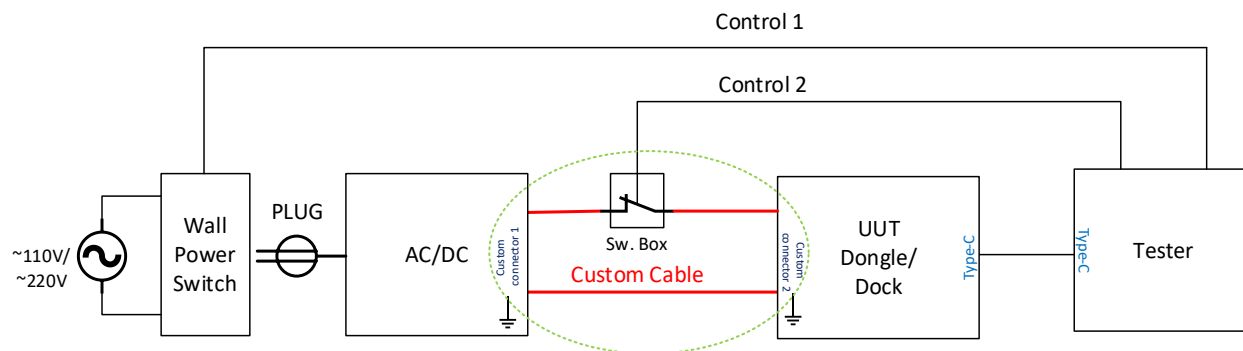


Figure 9 FRS Connection Diagram

The Tester Vendor shall provide Switch Boxes to accomplish these switching requirements. The UUT Vendor shall provide Custom Cables (see [Figure 13 Wiring diagram](#)) and the AC/DC adapter.

The circuitry of these boxes is at the discretion of the Test Equipment Vendor, except that the interface from the Tester shall be as described below, including connector type, electrical levels and logical sense of the input.

#### AC Mains / Wall Power Switch Box

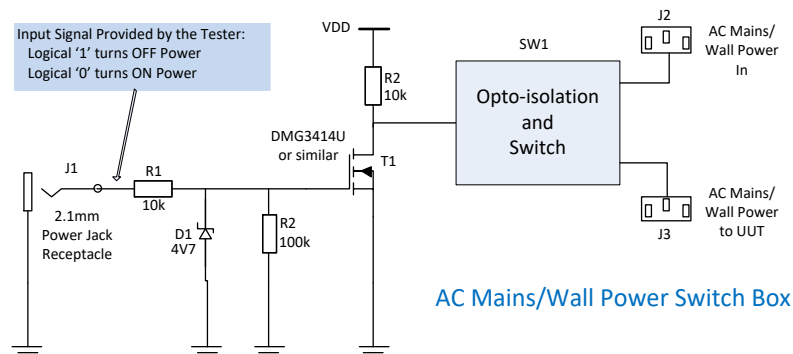


Figure 10 Schematic of the Switch Box for disconnecting AC Mains / Wall Power

Figure 10 Schematic of the Switch Box for disconnecting AC Mains / Wall Powershows the schematic of the Switch Box. The switch box has an AC connector (J2) to plug into the AC mains / wall power, and 2 additional receptacles:

- J1 is a 2.1mm barrel jack connector for receiving input signal from the Tester.
- J3 is an AC receptacle providing the UUT as its power supply.

The AC mains / wall power (the UUT's power supply) is allowed to pass if the Tester provides a logical "0" input to the switch box. Conversely, the AC mains / wall power is disconnected if the Tester provides a logical "1" as the input of the switch box, or if the cable from the Tester is not

connected. The function of each component in the schematic as shown in Figure 11 schematic of the standard switch box for disconnecting DC power:

- R1 is a 10 kOhm resistor, providing current limitation.
- R2 is a 100 kOhm resistor, providing high impedance termination.
- R3 is a 10 kOhm pull up resistor.
- D1 is a Zener diode for protecting the T1 N-channel MOSFET gate-to-source against any high voltage inadvertently applied to J1.
- SW1 represents a voltage-controlled switch circuit that is capable of switching AC voltage from 90 to 265V AC. Please note that this switch must contain suitable isolation circuitry to provide isolation of 2500V AC between the input and the switched AC voltage for safety purposes.
- T1 is an N-channel MOSFET that provides input signal to the switch SW1.

When the input signal is higher than 1.26V, the switch box is presented with a logical “1”:

- The T1 transistor is turned on.
- This in turn pulls the drain terminal of T1 transistor to zero and turns off the switch SW1.

When the input signal is lower than 0.54V or J1 is not connected, the switch box is presented with a logical “0”:

- The T1 transistor is turned off.
- This in turn pulls up the drain terminal of T1 transistor to VDD voltage and turns on the switch SW1.
- This allows the AC power to pass through the switch box.

## DC Switch Box

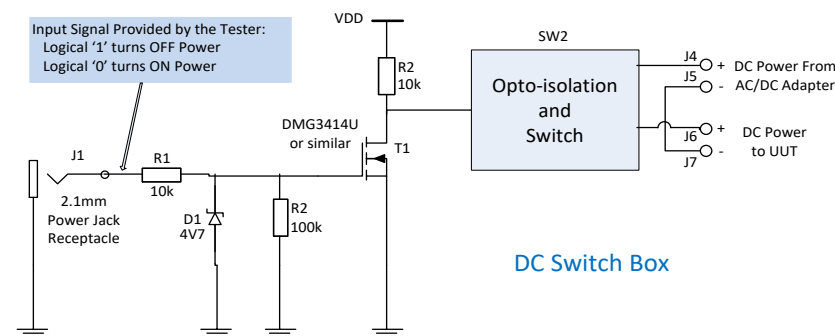


Figure 11 schematic of the standard switch box for disconnecting DC power

Most of the operation of the DC Switch Box (**Figure 12** Plug and Socket for disconnecting DC power) duplicates the AC mains / wall power Switch Box, however connectors J4 to J7 are provided to switch DC power between the UUT and its AC/DC Adapter. To accommodate different connectivity the connectors are 4mm test terminals, which will accept either 4mm test plugs or bare-ended wires. During Compliance Testing, the UUT Vendor is expected to provide suitable cables connected to the UUT and to its AC/DC Adapter.

A box for interrupting the DC power will need to be able to switch 15A. The connectors would have to carry this current too and be widely available. The 4mm test socket proposed will carry 15A, and the mating plugs are available in a shrouded version to protect the UUT Vendor's AC/DC Adapter from accidental shorting.



Figure 12 Plug and Socket for disconnecting DC power

If the UUT Vendor prefers bare ended cables, they will also be compatible and able to provide up to 15A.

Operating the DC Switch shall have the effect of disconnecting any normally user-accessible plugged DC supply connection.

## Cable interfaces and BOM

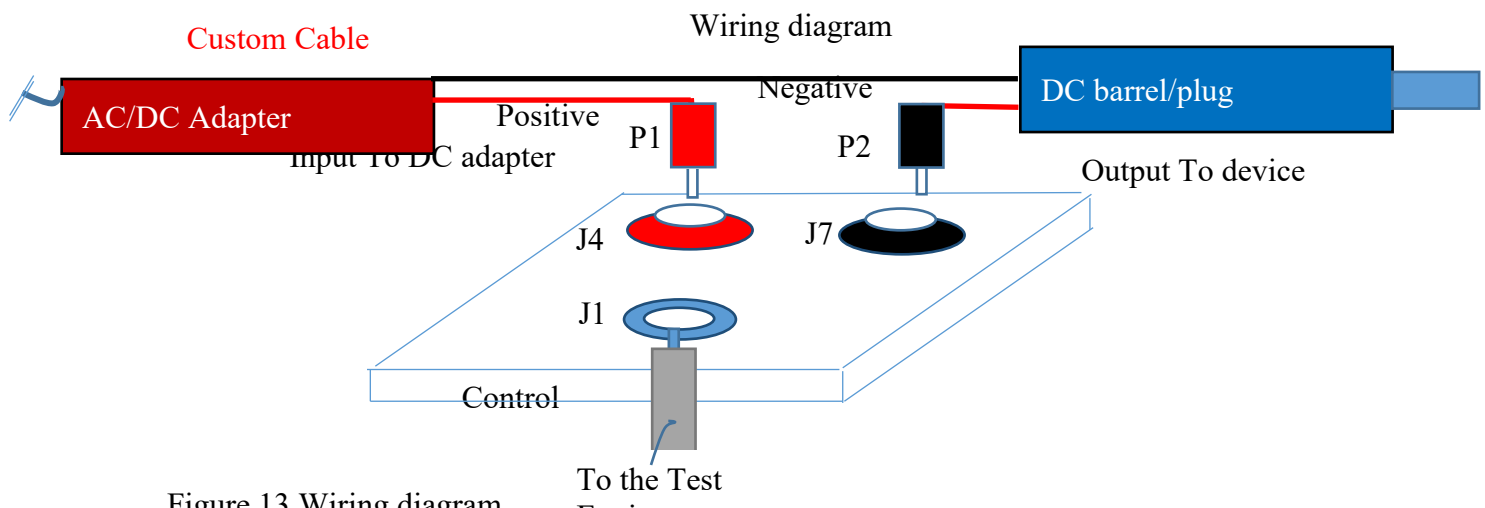


Figure 13 Wiring diagram

Table 18 Components

Component	Manufact.	Part Number	Newark #	Description
<b>P1</b>	Pomona	1325-2	30F1407	Conn banana Plug 15A, 60V Red
<b>P2</b>	Pomona	1325-0	30F1430	Conn banana Plug, 15A, 60V Black
<b>P3</b>	Switchcraft conxall	762Z	80AH1412	DC Power Connector
<b>J1</b>	Switchcraft Conxall	721A	49AC1396	DC Power Connector, 2.1mm Jack
<b>J4, J7</b>	2269-0	2269-0	30F1419	Banana Test Connector, 15A 25kV

## Appendix D: $V_{BUS}$ measurements

To give consistent results on measurements of  $V_{BUS}$  voltage it has been decided that all compliance test equipments **shall** use the same filtering when evaluating the voltage vs. the test spec.

This filter has been agreed to be a second order Butterworth low pass filter with a 3 dB cut off frequency of 15kHz. This is compatible with the parameters of all current testers.

The CTS proposes an implementation of the filter, first the function is listed, after that the coefficients are given.

### D.1 Filter Function

The below function is used with the coefficients listed in the Coefficients section to calculate the filtered  $V_{BUS}$  voltage.

When called with a sample value, the function returns the filtered sample value. It is intended to be called for all samples in the capture from the first to the last sample.

```
float Iir(float newSample)
{
    define nCoef 2

    static double y[nCoef + 1]; //output samples
    static double x[nCoef + 1]; //input samples
    int n;

    //shift the old samples
    for (n = nCoef; n > 0; n--) {
        x[n] = x[n - 1];
        y[n] = y[n - 1];
    }
}
```

```

//Calculate the new output
x[0] = newSample;
y[0] = m_aCoef[0] * x[0];
for (n = 1; n <= nCoef; n++)
    y[0] += m_aCoef[n] * x[n] + m_bCoef[n] * y[n];

return (float) y[0];
}

```

## D.2 Coefficients

Coefficients have been calculated for a number of sample rates.

Please note that these are informational and implementors are required to verify their implementations against the specification of the filter.

### 1.2.1 20μs sample interval

```

aCoef[0] = 0.39133577250176854178675966977607
aCoef[1] = 0.78267154500353708357351933955215
aCoef[2] = 0.39133577250176854178675966977607
bCoef[0] = 1.00000000000000000000000000000000
bCoef[1] = -0.36952737735124108198192516283598
bCoef[2] = -0.19581571265583302965396228501049

```

### 1.2.2 15μs sample interval

```

aCoef[0] = 0.24833913797217702090769364531297
aCoef[1] = 0.49667827594435404181538729062595
aCoef[2] = 0.24833913797217702090769364531297
bCoef[0] = 1.00000000000000000000000000000000
bCoef[1] = 0.18422210917344900016345832227671
bCoef[2] = -0.17757866106215708379423290352861

```

### 1.2.3 10μs sample interval

```

aCoef[0] = 0.13110643991662596086200665013166
aCoef[1] = 0.26221287983325192172401330026332
aCoef[2] = 0.13110643991662596086200665013166
bCoef[0] = 1.00000000000000000000000000000000

```



bCoef[1] = 0.74778917825850343970017775063752  
bCoef[2] = -0.27221493792500722763705311990634

#### 1.2.4 2μs sample interval

aCoef[0] = 0.00782020803349719149932806061543  
aCoef[1] = 0.01564041606699438299865612123085  
aCoef[2] = 0.00782020803349719149932806061543  
bCoef[0] = 1.00000000000000000000000000000000  
bCoef[1] = 1.73472576880927498166329314699396  
bCoef[2] = -0.76600660094326389337737737150746

#### 1.2.5 1μs sample interval

aCoef[0] = 0.00208056713549229247914307272538  
aCoef[1] = 0.00416113427098458495828614545076  
aCoef[2] = 0.00208056713549229247914307272538  
bCoef[0] = 1.00000000000000000000000000000000  
bCoef[1] = 1.86689227971171445297215996106388  
bCoef[2] = -0.87521454825368372176797038264340

## Appendix F: Timing table

Table 19 Timing Table & Calculations

Timer	Spec Rev.	min	max	Reference
<i>tSenderResponse</i>	PD2	24 ms	30 ms*	[PD2 Spec]
	PD3	27 ms	50 ms	[PD3 Spec]
<i>tCtsBIST</i>		-	15 ms	
<i>tCtsChunkMargin</i>		-	5 ms	
<i>tCtsDataResetDiscovEntry</i>		-	2s	
<i>tCtsErrorRecovery</i>		-	15 ms	Used in tests for timing the Error Recovery
<i>tCtsGoodCRC</i>		-	10 ms	
<i>tCtsHardReset</i>		-	15 ms	
<i>tCtsInfinite</i>		-	5s	Arbitrary, relatively long number, making sure that UUT

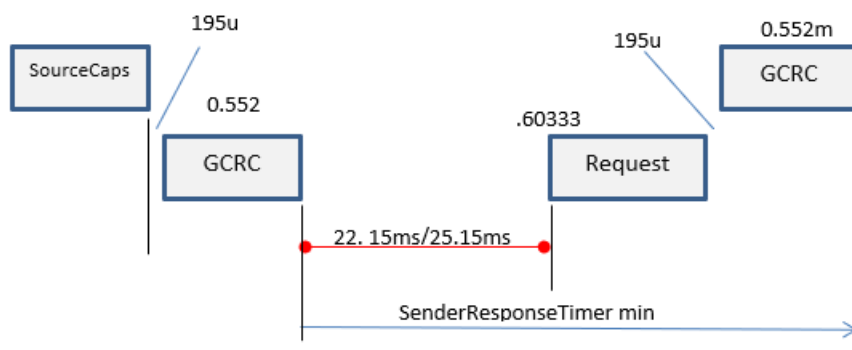
				does a certain action
<i>tCtsAfterLastPoint</i>		-	500 ms	
<i>tCtsSnkRequest</i>	PD2	-	22.26 ms	Figure 13 Wiring diagram
	PD3	-	25.26 ms	Figure 13 Wiring diagram
<i>tCtsSrcAccept</i>	PD2	-	22.76 ms	Figure 14 tCtsSnkRequest Calculations
	PD3	-	25.76 ms	Figure 14 tCtsSnkRequest Calculations Figure 15 tCtsSrcAccept Calculation
<i>tCtsSenderResponse</i>		-	50 ms	
<i>tCtsWait</i>		-	1 Sec	
<i>tCtsPSSourceOnDeadline</i>	PD3	$tPSSourceOn.min - (2 * tReceive)$	$tPSSourceOn.min - tReceive$	
<i>tCtsPSSourceOffDeadline</i>	PD3	$tPSSourceOn.min - (2 * tReceive)$	$tPSSourceOn.min - tReceive$	
<i>nCtsWaitCount</i>			20	

\*The tester uses PD3 max timing on PD3 devices operating in PD2 mode.

#### tCtsSnkRequest Calculations

tCtsSrcAccept is timed from the end of the GoodCRC sent by the Tester in response to the Source Capabilities, until the start of the Request sent by the Tester. These times allow 0.5ms for a margin of error for the tester.

Figure 14 tCtsSnkRequest Calculations



**Duration of Accept Message (sent by tester at nominal bit rate)**

- Preamble 64 bits
- Start of Packet 20 bits
- Message Header 20 bits
- CRC 40 bits
- End of Packet 5 bits

Total bits = 149

Duration of bit at 300kb/s = 3.333e-6

Duration of message = 496.62e-6

**Duration of Longest GoodCRC Message**

- Preamble 64 bits
- Start of Packet 20 bits
- Message Header 20 bits
- CRC 40 bits
- End of Packet 5 bits

Total bits = 149

Duration of bit at 270kb/s = 3.704e-6

Duration of message = 0.552 ms

Latest valid time to start sending Request PD2

$$24 - (0.49662 + 0.195 + 0.552) =$$

Latest valid time to start send 22.76ms

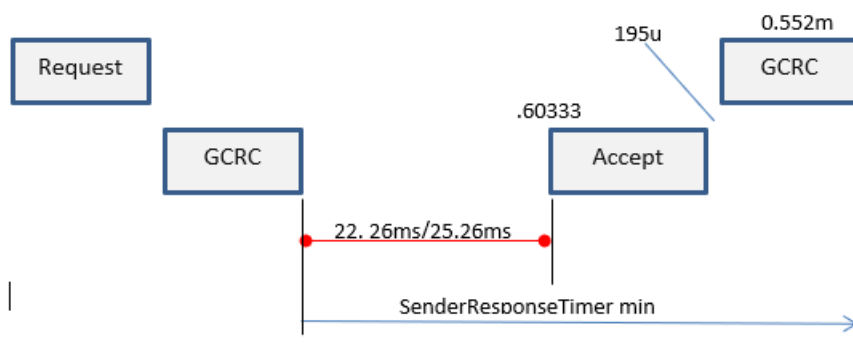
$$27 - (0.60333 + 0.195 + 0.552) =$$

25.76ms

**tCtsSrcAccept Calculations**

tCtsSrcAccept is timed from the end of the GoodCRC sent by the Tester in response to the Request, until the start of the Accept sent by the Tester. These times allow 0.5ms for a margin of error for the tester.

Figure 15 tCtsSrcAccept Calculation

**Duration of Request Message (sent by tester at nominal bit rate)**

- Preamble 64 bits
- Start of Packet 20 bits
- Message Header 20 bits
- Data bits 32 bits
- CRC 40 bits
- End of Packet 5 bits

Total bits = 181

Duration of bit at 300kb/s = 3.333e-6  
Duration of message = 603.33e-6

#### Duration of Longest GoodCRC Message

- |                   |         |
|-------------------|---------|
| • Preamble        | 64 bits |
| • Start of Packet | 20 bits |
| • Message Header  | 20 bits |
| • CRC             | 40 bits |
| • End of Packet   | 5 bits  |

Total bits = 149

Duration of bit at 270kb/s = 3.704e-6

Duration of message = 0.552 ms

Latest valid time to start sending Request PD2

$$24 - (0.60333 + 0.195 + 0.552) =$$

Latest valid time to start send 22.65ms

$$27 - (0.60333 + 0.195 + 0.552) =$$

25.65ms

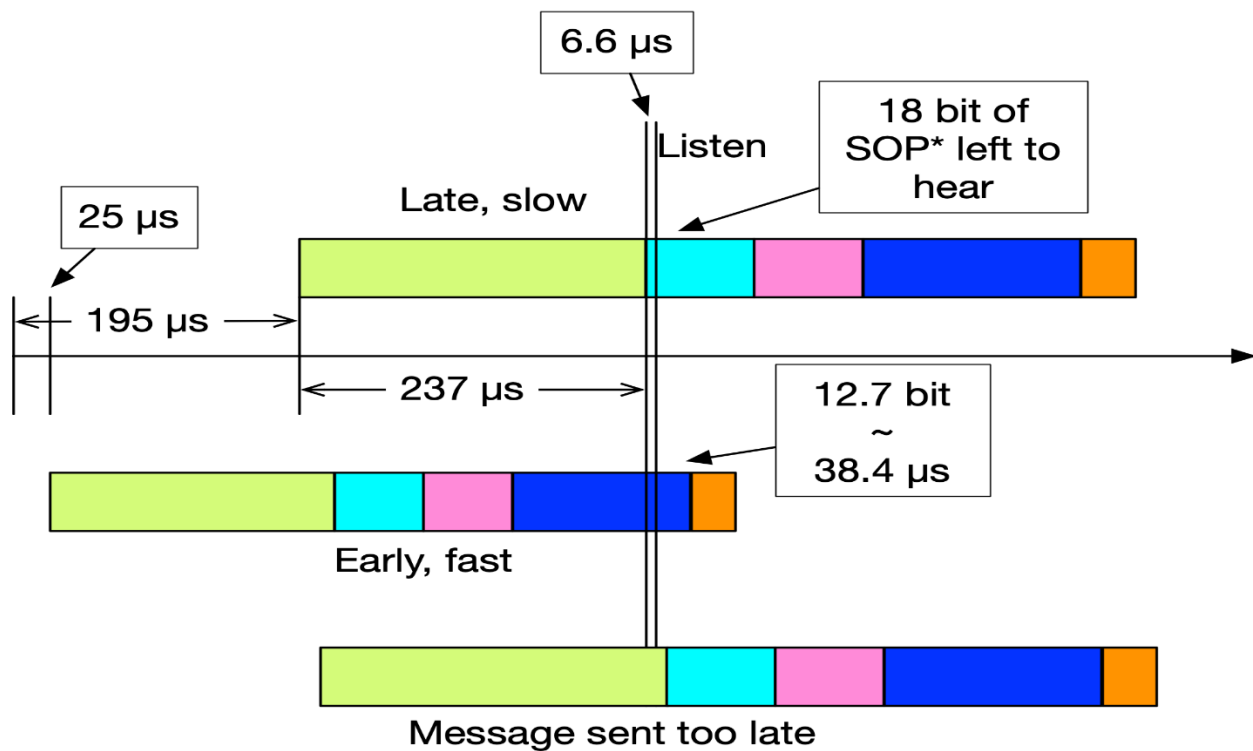
## Appendix E. Slew Rate Measurements.

To have consistent results on signal's slew rate measurements the Tester shall take the following steps:

- Turn all filtering and sample averaging off and work only with the raw signal samples.
- Collect data samples with rate of at least 20uS/Sample,
- Start collecting signal samples after receiving *Accept* Message and before sending the last bit of *GoodCRC* Message. Record the Voltage Level as  $V_1$ .
- Stop collecting signal samples upon receiving *PS\_RDY* Message. Record the Voltage Level as  $V_2$ .
- Calculate the difference  $\Delta V = |V_2 - V_1|$
- If  $\Delta V < 3.5V$ , the slew rate measurement could not be accurately calculated, and the measurement disregarded.
- Find the first sample that exceeds  $V_1 + 0.1 * \Delta V$  for the Positive Transition ( $V_2 > V_1$ ) or getting below  $V_2 + 0.9 * \Delta V$  for the Negative Transitions ( $V_2 < V_1$ ) and corresponding sample number  $S_{n\_A}$  from the last bit of *GoodCRC* – point **A**
- Find the first sample that exceeds  $V_1 + 0.9 * \Delta V$  for the Positive Transitions or getting below  $V_2 + 0.1 * \Delta V$  for the Negative Transitions and corresponding sample number  $S_{n\_B}$  from the last bit of *GoodCRC* – point **B**
- Find  $SR = (V_{point\_B} - V_{point\_A}) / (S_{n\_B} - S_{n\_A}) * R$

## Appendix F. Signal In Presence of Noise

Figure 16 Signal Timing in Presence of Noise



#### Legend:

<span style="display: inline-block; width: 20px; height: 15px; background-color: #90EE90; border: 1px solid black;"></span>	Preamble
<span style="display: inline-block; width: 20px; height: 15px; background-color: #00FFFF; border: 1px solid black;"></span>	SOP*
<span style="display: inline-block; width: 20px; height: 15px; background-color: #FF69B4; border: 1px solid black;"></span>	Message - GoodCRC
<span style="display: inline-block; width: 20px; height: 15px; background-color: #0000FF; border: 1px solid black;"></span>	CRC
<span style="display: inline-block; width: 20px; height: 15px; background-color: #FF8C00; border: 1px solid black;"></span>	EOP

#### Timing details

- 237  $\mu$ s is the longest time required to send preamble
- 6.6  $\mu$ s is a margin to ensure not seeing SOP.
- Message could be over in 25 $\mu$ s + 149 bits at fastest speed = 472  $\mu$ s (leaves minimum 38.4 $\mu$ s or 12.7 bits)

### Appendix G Managed Capability Port Test Flow

Below flow chart depicts the test flow for *Managed Capability Port* DUTs with the best effort sequence to derive the maximum *Source Capabilities* from the DUT. If the PDP of the *Source Capabilities* is less than the maximum PDP in the VIF, additional test conditions would apply to evaluate the *Source Capabilities* per the relevant VIF or *Source\_Info* parameters.

