

USB Power Delivery Compliance Test Specification

Revision: ***1.4 V5 OR***

Release date: ***November 2022***

LIMITED COPYRIGHT LICENSE

THE USB 3.0 PROMOTERS GRANT A CONDITIONAL COPYRIGHT LICENSE UNDER THE COPYRIGHTS EMBODIED IN THE USB POWER DELIVERY SPECIFICATION TO USE AND REPRODUCE THE SPECIFICATION FOR THE SOLE PURPOSE OF, AND SOLELY TO THE EXTENT NECESSARY FOR, EVALUATING WHETHER TO IMPLEMENT THE SPECIFICATION IN PRODUCTS THAT WOULD COMPLY WITH THE SPECIFICATION. WITHOUT LIMITING THE FOREGOING, USE THE OF SPECIFICATION FOR THE PURPOSE OF FILING OR MODIFYING ANY PATENT APPLICATION TO TARGET THE SPECIFICATION OR USB COMPLIANT PRODUCTS IS NOT AUTHORIZED. EXCEPT FOR THIS EXPRESS COPYRIGHT LICENSE, NO OTHER RIGHTS OR LICENSES ARE GRANTED, INCLUDING WITHOUT LIMITATION ANY PATENT LICENSES. IN ORDER TO OBTAIN ANY ADDITIONALLY INTELLECTUAL PROPERTY LICENSES OR LICENSING COMMITMENTS ASSOCIATED WITH THE SPECIFICATION A PARTY MUST EXECUTE THE USB 3.0 ADOPTERS AGREEMENT. NOTE: BY USING THE SPECIFICATION, YOU ACCEPT THESE LICENSE TERMS ON YOUR OWN BEHALF AND, IN THE CASE WHERE YOU ARE DOING THIS AS AN EMPLOYEE, ON BEHALF OF YOUR EMPLOYER.

INTELLECTUAL PROPERTY DISCLAIMER

THIS SPECIFICATION IS PROVIDED TO YOU “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OR IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. THE PROVISION OF THIS SPECIFICATION TO YOU DOES NOT PROVIDE YOU WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS.

Please send comments via electronic mail to techsup@usb.org

For industry information, refer to the USB Implementers Forum web page at <http://www.usb.org>

USB Type-C® and USB4™ are trademarks of the Universal Serial Bus Implementers Forum (USB-IF). Thunderbolt™ is a trademark of Intel Corporation.

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

Copyright © 2010-2022, USB 3.0 Promoter Group: Apple Inc., Hewlett-Packard Inc., Intel Corporation, Microsoft Corporation, Renesas, STMicroelectronics, and Texas Instruments. All rights reserved.

Editors

Dmitriy Berchanskiy
Bob Dunstan

Intel Corporation
Renesas Electronics Corp.

Contributors

Abel Astley	Ellisys
Nathalie Ballot	STMicroelectronics
Dmitriy Berchanskiy	Intel Corporation
Naresh Botsa	Granite River Labs
Sten Carlsen	MQP Electronics Ltd.
Pat Crowe	MQP Electronics Ltd.
Harshavardhan Reddy Gonugunta	Granite River Labs
Tyler Joe	Teledyne LeCroy
Diane Lenox	USB Implementers Forum
Balamurugan Manialagan	Granite River Labs
Tim McKee	Intel Corporation
Mike Micheletti	Teledyne LeCroy
Prasannakumar Marikunte Nagaraju	Granite River Labs
Mahesh Nakala	Granite River Labs
Krishnamurthy Parameshwara	Granite River Labs
Mario Pasquali	Ellisys
Hao Sun	Ellisys
Chuck Trefts	Ellisys
Rajaraman Venkatachalam	Granite River Labs
Deric Waters	Texas Instruments
Biju Erayamkot Panayamthatta	Texas Instruments
Shafiuddin Mohammed	Texas Instruments
Tim Wei	Ellisys

Table of Contents

INTELLECTUAL PROPERTY DISCLAIMER	2
Editors	3
Contributors	3
Revision History	8
1 Introduction	9
1.1 Overview	9
1.2 Terms and Abbreviations	10
1.2.1 The types of UUTs	10
1.3 Keywords.....	11
1.3.1 Check.....	11
1.3.2 PD2 mode.....	11
1.3.3 PD3 mode.....	11
1.3.4 Revision Alignment	11
1.4 Test Requirements	11
1.4.1 VCONN Levels for Testing Cable Markers or VPDs	11
1.4.2 Test Condition Codes	12
1.4.3 “Supports Structured VDMs”	15
1.4.4 Error Recovery	15
1.4.5 V_{BUS} Measurement – Point of Reference	16
1.4.6 Tester Default Behavior	16
1.4.7 Reporting Missing Message Bytes	16
1.4.8 Reporting When Test Cannot be Completed	16
2 Preparation Procedure	16
2.1 PD3 Mode	16
TEST.PREP.PR.1 Preparation for Bring-up Source UUT	16
3 Common Checks, Bring-up Procedures, and Common Procedures.....	17
3.1 Common Checks.....	17
3.1.1 PD2 and PD3 Modes.....	17
3.1.2 PD3 Mode	32
3.2 Bring-up Procedures	37
3.2.1 PD2 and PD3 Modes.....	37
3.3 Common Procedures	51

3.3.1	PD2 and PD3 Modes.....	51
3.3.2	PD3 Mode.....	61
4	Physical Layer Specific Tests	67
4.1	Test Procedures for All UUTs	71
4.1.1	PD3 Mode.....	71
4.2	Test Procedures for Source and Sink Capable UUTs.....	87
4.2.1	PD3 Mode.....	87
5	Protocol Specific Tests	90
5.1	Test Procedures for All UUTs	90
5.1.1	PD2 and PD3 Modes.....	90
5.1.2	PD3 Mode.....	103
5.2	Test Procedures for Source and Sink Capable UUTs.....	119
5.2.1	PD2 and PD3 Modes.....	119
5.2.2	PD3 Mode.....	119
5.3	Test Procedures for Source Capable UUT	129
5.3.1	PD2 and PD3 Modes.....	129
5.3.2	PD3 Mode.....	141
5.4	Test Procedures for Sink Capable UUT	152
5.4.1	PD2 and PD3 Modes.....	152
5.4.2	PD3 Mode.....	166
5.5	Structured VDM Test Procedures for Source Capable UUT.....	172
5.5.1	PD2 and PD3 Modes.....	172
5.6	Structured VDM Test Procedures for Sink Capable UUT	175
5.6.1	PD2 and PD3 Modes.....	175
5.7	Structured VDM Test Procedures for Cable Plug UUT.....	184
5.7.1	PD2 and PD3 Modes.....	184
5.7.2	PD3 Mode.....	189
6	Power Supply Specific Tests.....	191
6.1	Test Procedures for Source Capable UUT.....	193
6.1.1	PD2 and PD3 Modes.....	193
6.2	Test Procedures for Sink Capable UUT	205
6.2.1	PD2 and PD3 Modes.....	205
7	Extended Power Range Tests.....	216

7.1 Test Procedure for EPR Source Capable UUT	216
7.1.1 PD3 Mode	216
7.1.2 PD3 Mode – EPR_Power_Supply Specific Tests	235
7.2 Test Procedure for EPR Sink Capable UUT	240
7.2.1 PD3 Mode	240
8 Fast Role Swap Tests	251
8.1 Fast Role Swap Tests – Initial Source	251
8.1.1 PD3 Mode	251
8.2 Fast Role Swap Tests – Initial Sink	260
8.2.1 PD3 Mode	260
9. USB4	268
9.1 USB4 Tests – Data Reset	268
9.1.1 PD3 Mode	268
9.2 USB4 Tests – Enter USB4.....	277
9.2.1 PD3 Mode	277
9.3 USB4 Tests – Cable Response	286
9.3.1 PD3 Mode	286
Appendix A: Requirements for Compliance Test Results	288
Appendix B: XML Output Example.....	289
Appendix C: Power Disconnection for FRS Initial Source Testing.....	292
AC Mains / Wall Power Switch Box.....	293
DC Switch Box	294
Cable interfaces and BOM	295
Appendix D: V_{BUS} measurements	296
D.1 Filter Function.....	296
D.2 Coefficients.....	297
Appendix F: Timing table	298
tCtsSnkRequest Calculations.....	298
tCtsSrcAccept Calculations.....	300

TABLES

Table 1 UUT Types	10
Table 2 Cable Types	10

Table 3 - VCONN Test Voltages	11
Table 4 Test Conditions Codes for Cables or VPDs	15
Table 5 Max_Current_Field Truth Table	26
Table 6 Src_PDO_Max_PowerN Truth Table	26
Table 7 Receiver Test Signaling Table	67
Table 8 Rev 2 Port - Number of VDOs Expected	178
Table 9 Rev 2 Cable - Number of VDOs Expected	178
Table 10 Rev 3 Port Number of VDOs Expected	178
Table 11 Cable - Number of VDOs Expected.....	179
Table 12 PDO Sequence	191
Table 13 EPR PDO Sequence	191
Table 14 Sink Capabilities Extended Message	198
Table 14 Components	295
Table 15 Timing Table & Calculations	298
Figure 1: using mid-level to find the reference unit interval.....	67
Figure 2: Example of producing eye diagram	68
Figure 3: Noise injection generator and calibration fixture setup.....	68
Figure 4: Two-tone noise calibration	69
Figure 5: AWG measurement waveforms.....	70
Figure 6 Voltage and Current Tolerances	198
Figure 7 FRS Testing Concept.....	252
Figure 8 Schematic of the Switch Box for disconnecting AC Mains / Wall Power	293
Figure 9 schematic of the standard switch box for disconnecting DC power	294
Figure 10 Wiring diagram.....	295

Revision History

Revision	Release Date	Description
1.0	September 15, 2019	Initial release
1.0	November 22, 2019	Adds missing cover page, copyright notice and table of contents. No change to the compliance tests.
1.2	September 19, 2020	Initial Merged specification
1.2 V2	June 20, 2021	With all fixes from workshop
1.3	June 21, 2021	Adds FRS, Captive Cables, EPR SRC
1.4	October 3, 2021	Adds EPR SNK, USB4
1.4 V2	January 16, 2022	November 2021 workshop fixes
1.4 V3	April 22, 2022	Major changes for COMMON.CHECK.PD.7
1.4 V4	July 17, 2022	Common.Proc.PD.7, DiscovSOP_Resp_Chk, tSenderResponse
1.4 V5	November 11, 2022	TEST.PD.PROT.PORT3_UPDATES Missing reference to vPpsNew SOURCE INFO GET_REVISION_INFO PD.PROT.ALL.2 tHardReset COMMON.CHECK.PD.10 Check Message Header COMMON.CHECK.PD.2 Check Extended Message COMMON.PROC.PD3.2 UUT Sent EPR Source Cap message Declare EPR Mode Capable bit setting GotoMin EPR Capability bit removal tSenderResponse Enter USB Request Ensuring Capability Offered

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 followed:

- 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 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.

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 revision interoperability check has been done, as described in 6.2.1.5 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 2 shows the complete list of Test Conditions used for Ports.

Table 2 - 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 4 *Test Conditions Codes for Cables or VPDs* shows the complete list of Test Conditions used for Cable Markers or VPDs.

Table 4 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.

1.4.5 V_{BUS} Measurement – Point of Reference

For a captive cable, V_{BUS} 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 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'.

2 Preparation Procedure

2.1 PD3 Mode

The preparation procedures in this subsection are applicable when the test is running in PD3 mode before all other tests on any UUT capable of being a Source i.e., provider only, Provider-Consumer, DRP and Consumer-Provider, to ensure that shared multiport UUTs behave correctly in their test mode. After running this once, all ports of the UUT may be tested, provided power is not removed from the UUT.

TEST.PREP.PR.1 Preparation for Bring-up Source UUT

Description: The Tester checks whether the UUT needs to be set into Shared Test Mode prior to all other testing.

Test Conditions:

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

Test Specific Tester Behavior: N/A

Assumes that the Test Cable is initially connected to the Port under Test.

1. Connect the UUT to its power source.
2. If VIF field Port_Source_Power_Type is not set to Shared, then the preparation procedure is complete.
3. Otherwise, if the VIF field Master_Port is set to YES, the Tester sends a BIST Shared Test Mode Entry message to the UUT, and the preparation procedure is complete.
4. If the VIF field Master_Port is set to NO, the Tester uses the VIF to locate a component which has the VIF field Master_Port parameter is set to YES and requests the operator to connect the Test Cable to that Port on the UUT. If one is not found, the check fails [TEST.PREP.PR.1#1]
5. When this has been done, the Tester sends a BIST Shared Mode Test Entry message to the UUT.
6. The Tester requests the operator to connect the Test Cable to the Port under Test.
7. When this has been done, the preparation procedure is complete.

TEST NOTES:

- Once the UUT is placed in the test mode it is expected to remain in the test mode till it is powered down.
- The test procedure should be performed only one time per port.

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]

- For *Source_Capabilities* and *Discover Identity* Commands before the first response from the Tester after detaching, *Hard Reset* or Error Recovery
 - Bit 15 (Extended for PD3). The check fails if this field is not set to 0b
 - Bit 7...6 (Specification Revision). The check fails if this field is not set to 10b. If the message is a *Discover Identity* over SOP' and it is not the first time the message has been sent after the detach, *Hard Reset*, or Error Recovery, this check does not apply.
 - Bit 4...0 (Message Type). The check fails if the Message Type is not valid
- All messages
 - Bits 7...6 (Specification Revision). The check fails if this field is set to either 00b or 11b.
- Bits 14...12: Number of Data Objects
 - 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
 - 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
 - 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
 - Data messages:
 - Request* Message: Number of Data Objects = 1
 - Battery_Status message: Number of Data Objects = 1
 - Alert* Message: Number of Data Objects = 1
 - Get_Country_Info message: Number of Data Objects = 1
 - Enter_USB* Message: Number of Data Objects = 1
 - EPR_Request* Message: Number of Data Objects = 2
- Bits 11...9: *MessageID*. The check fails if the value is not 000b under the following conditions:
 - The first message on each SOP* type after a *Hard Reset* was sent or received
 - The message is a *Soft Reset*
 - The first message on the same SOP* type after receiving Soft_Reset
 - The first message on SOP after a successful Power Role Swap
 - 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*
- 5. Bits 11...9: *MessageID*. If *MessageID* is incremented for a valid retry scenario as described in the test procedures, fail the check. (On retry the message all fields should remain the same from the previously received message.) Else, fail the check if *MessageID* is not incremented.
- 6. 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
- 7. 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. For a *Ping* or *GotoMin* Message, the check fails if Port Power Role is not Source
 - c. For a *Request*, 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
- 8. 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
- 9. Bit 5: Port Data Role (SOP packet type only). The check fails if Port Data Role does not represent the UUT port data role
- 10. 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_{BUS} when behaving as a Source or performs an unexpected Error Recovery. [COMMON.CHECK.PD.5#5].

Unless specifically mentioned, the check fails if *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].

Unless specifically mentioned, the check fails if *GotoMin* Message is sent by the UUT [COMMON.CHECK.PD.5#3]. The Tester does not set the GiveBack flag during a test.

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)
- *Ping* (when the UUT is not in *PE_SRC_Ready*)

In PD3 Mode:

- *GoodCRC*
- BIST
- *Sink_Capabilities*
- *Accept*
- *Reject*
- *Wait*
- *PS_RDY*
- Structured VDM response (ACK, NAK, BUSY)
- *Ping* (when the UUT is not in *PE_SRC_Ready*)
- *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*, *GoToMin*, *Accept*, *Reject*, *Ping*, *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, a 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 irregardless 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 – run all checks except those marked '**FULL CAPABILITY ONLY**'

NOTE

: If VIF Parameter Has_Invariant_PDOs is set to N, the checks marked '**FULL CAPABILITY ONLY**' will be checked by the procedures in:

TEST.PD.PS.SRC.1 Multiple Request Load Test ,

TEST.PD.PS.SRC.2 PDO Transitions,

TEST.PD.PS.SNK.2 Initial Sink PDO Transitions Post PR Swap ,

TEST.PD.PS.SNK.3 Multiple Request Load Test Post PR Swap

Check Applicability: *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 [COMMON.CHECK.PD.7#1]
 - a. For source, currents greater than 3A and/or voltages greater than 20V.
 - b. For sink with DRP bit is set, if this message is response of *Get_Source_Cap* then tester checks whether DUT responds with its maximum capability

2. For sink, the check fails if UUT initiates source capability message without receiving *Get_Source_Cap* Message from 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 (count of how many Src_PDO_Supply_TypeN is set to PPS).
4. For the first PDO, the Tester check 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. For *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
 - h. For *Source_Capabilities* Message after the Tester has sent *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. Bits 23 (EPR Mode Capable) matches VIF field EPR_Supported_as_Src in PD3 mode
 - iii. Bit 22 (Reserved) set to 0b if the test is running in PD3 mode
 - iv. Bits 24...22 (Reserved) set to 000b if the test is running in PD2 mode
 - i. B21...20 (Peak Current) matches VIF field Src_PDO_Peak_Current1
 - j. B19...10 (Voltage) set to 5V
 - k. B9...0 (Maximum Current)
 - i. For UUT as a Sink the current shall be Src_PDO_Max_Current1
 - ii. **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. 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 5** 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. **FULL CAPABILITY ONLY:** The Tester compares all PDOs to VIF field PD_Power_as_Source 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 (PD_Power_as_Source/Voltage) or RoundDown (PD_Power_as_Source/Voltage) to the nearest 10mA
 - ii. Check that all the normative voltages are supported by UUT based on the PD_Power_as_Source as per the Table 10-2 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 PD_Power_as_Source
 - c. SPR PPS APDO
 - i. Check that the maximum power is RoundDown (PD_Power_as_Source/Prog Voltage) to the nearest 50mA
 - ii. Check that all the normative SPR PPS voltages are supported by UUT based on the PD_Power_as_Source as per Table 10-7 in the PD3 specification
 - 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 Table 10-8 in the PD3 specification.
8. **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:
 1. For UUT as a Sink the current shall be Src_PDO_Max_CurrentN
 2. 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 5** Max_Current_Field Truth Table

 - c. If Variable Supply PDO:
 - i. Src_PDO_MinVoltageN

- ii. Src_PDO_MaxVoltageN
- iii. Max_current_Field:
 - 1. For UUT as a Sink the current shall be Src_PDO_Max_CurrentN
 - 2. For UUT as a Source:
 - a. If Src_PDO_Max_currentN \leq 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 5](#) Max_Current_Field Truth Table

- d. If Battery Supply PDO:
 - i. Src_PDO_MinVoltageN
 - ii. Src_PDO_MaxVoltageN
 - iii. Src_PDO_Max_PowerN field:
 - 1. For UUT as a Sink the current shall be SrcPDO_Max_PowerN
 - 2. 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 6](#) Src_PDO_Max_PowerN Truth Table

- e. If Programmable Power Supply PDO:
 - i. Src_PDO_MinVoltageN
 - ii. Src_PDO_MaxVoltageN
 - iii. Max_current_field:
 - 1. For UUT as a Sink the current shall be Src_PDO_Max_CurrentN
 - 2. For UUT as a Source:
 - a. If Src_PDO_Max_currentN \leq 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 5** Max_Current_Field Truth Table

9. 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]
10. The Tester checks that no Fixed PDO has the same voltage as any other [COMMON.CHECK.PD.7#10]
11. The Tester checks that no Variable PDO has the same voltage range as any other [COMMON.CHECK.PD.7#11]
12. The Tester checks that no Battery PDO has the same voltage range as any other [COMMON.CHECK.PD.7#12]

Table 5 Max_Current_Field Truth Table

UUT is Source	Yes				No
Src_PDO_Max_CurrentN in VIF	<= 3A	>3A			any
Emulated eMarker	3A/5A/None	3A or None	-	5A	3A/5A/None
UUT has Captive Cable	Don't care	No	Yes	No	Don't care
Expected Message PDO Max Current for check	SrcPDO_Max_currentN	3A	SrcPDO_Max_currentN		SrcPDO_Max_currentN

Table 6 Src_PDO_Max_PowerN Truth Table

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

COMMON.CHECK.PD.8 Check Request Message

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

Check Applicability: *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), matches the VIF field "EPR_Supported_As_Snk"
 3. Bits 21 and 20 (Reserved) set to 000b if the test is running in PD3 mode

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. If B27 (GiveBack flag) = 0b
 - 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. If B27 (GiveBack flag) = 1b
B9...0 (Minimum Operating Current) does not exceed B19...10 (Operating Current)
3. Battery Request Data Object
 - a. B19...10 (Operating Power) does not exceed the power offered by the referenced PDO
 - b. If B27 (GiveBack flag) = 0b
 - 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. If B27 (GiveBack flag) = 1b
B9...0 (Minimum Operating Power) does not exceed B19...10 (Operating Power)
4. Programmable Power Supply Request
 - a. B6...0 (Operating Current) is less than or equal to the Maximum Current offered by the referenced PDO

- b. B19...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 (reserved):
 - a. Set to 00b if Command = 0...15
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 *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 *Request* or *Source_Capabilities* Message, or Unchunked Extended Messages Supported bit set to 0b (in UUT sent Request or *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: *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
 - 1. Bits 0...6 (Source's rated PDP) is set to 100
 - ii. if EPR_Supported_As_Src is not set
 - 1. Bits 0...6 (Source's rated PDP) is in consistent with PD_Power_As_Source
 - 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: *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)

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*

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: *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 *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...20 (Reserved) set to 000b if the test is running in PD3 mode
2. Fixed and Variable Request Data Object
 - a. B19...10 (Operating Current) does not exceed the current offered by the referenced PDO
 - b. If B27 (GiveBack flag) = 0b
 - 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. If B27 (GiveBack flag) = 1b

- B9...0 (Minimum Operating Current) does not exceed B19...10 (Operating Current)
3. Battery Request Data Object
 - a. B19...10 (Operating Power) does not exceed the power offered by the referenced PDO
 - b. If B27 (GiveBack flag) = 0b
 - 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. If B27 (GiveBack flag) = 1b
 - B9...0 (Minimum Operating Power) does not exceed B19...10 (Operating Power)
 4. 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. B19...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
 5. 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
 6. Tester checks that the PDO(PDO2) in the *EPR_Request* Message exactly matches the PDO in the latest *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_Request* Messages sent by the UUT.

Check Applicability: *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_PDO_PDP_Rating” VIF field if action filed 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.

Check Applicability: *EPR_Source_Capabilities* Message

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. 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. For *Source_Capabilities* Message after the Tester has sent *Request* Message
 1. Bit 24 (Unchunked Extended Messages Supported) matches VIF field Unchunked_Extended_Messages_Supported if the test is running in PD3 mode
 2. Bit23 (EPR Mode Capable) matches VIF field EPR_Supported_as_Src if the test is running in PD3 mode
 3. Bits 22 (Reserved) set to 00b if the test is running in PD3 mode
 - h. B21...20 (Peak Current) matches VIF field Src_PDO_Peak_Current1
 - i. B19...10 (Voltage) set to 5V
 - j. B9...0 Max_current_field
 1. For UUT as a Sink the current shall be Src_PDO_Max_Current1
 2. For UUT as a Source:
 - i. If Src_PDO_Max_current1 <= 3A, Src_PDO_Max_Current1
 - ii. If Src_PDO_Max_current1 > 3A:
 1. 3A if
 - a. emulated cable is not e-marked, or it is 3A capable only and
 - b. Captive_Cable = NO
 2. Otherwise, Src_PDO_Max_Current1

NOTE: For the truth table of **Max_Current_Field**, which is **Src_PDO_Max_current1**, see **Table 5** Max_Current_Field Truth Table

3. If it is not the first Fixed Supply PDO, check that B29...22 is set to zero
[COMMON.CHECK.PD3.3#3]
4. If it is a Programmable Power Supply APDO, check that B26...25, B16 and B7 are set to zero
[COMMON.CHECK.PD3.3#4]
5. The Tester compares all the EPR PDOs (position 8 to 11) to VIF field PD_Power_as_Source 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 PD_Power_as_Source as per the Table 10-9 in the PD3 specification
 3. Check that the Voltage is either 28V, 36V or 48V
 - b. EPR AVS APDO
 1. Check that PDP field matches with PD_Power_as_Source VIF field value
 2. Check that all the normative voltages are supported by UUT based on the PD_Power_as_Source as per the Table 10-9 in the PD3 specification
 3. Check that the Maximum Voltage is either 28V, 36V or 48V
 4. Check that the Minimum Voltage is 15V
 - 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.
6. 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
7. Check that the UUT follows all the PDO specific checks defined in COMMON.CHECK.PD.7 for all the SPR PDO in PDO position between 1 and 7. [COMMON.CHECK.PD3.3#7]

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: *Sink_Capabilities* Message

Perform the following checks on all *Sink_Capabilities* Messages: [COMMON.CHECK.PD3.4#1]

1. The Tester runs the following field checks on the *Sink_Capabilities* Message:
 - a. 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
 - 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
 - b. For each PDO, the Tester checks:
 - i. The Tester checks consistency of Snk_PDO_Supply_TypeN
 - ii. If Fixed Supply PDO:
 1. In PD3, B22...20 (Reserved) are set to zero
 2. The Tester checks consistency of Snk_PDO_VoltageN
 3. 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
 - vi. If Adjustable Voltage Supply APDO:
 1. B29...28 set to 01b
 2. B27...26 (reserved) set to zero
 3. B16 (Reserved) set to zero
 4. The Tester checks consistency of Snk_PDO_Max_VoltageN

5. The Tester checks consistency of Snk_PDO_Min_VoltageN
6. The Tester checks if the PDP is consistent with "Snk_PDO_PDP_Rating" VIF field.
- c. The Tester compares all PDOs to VIF field PD_Power_as_Sink and check that they meet the requirements of the Power Rules
- d. 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
- e. The Tester checks that no Fixed PDO has the same voltage as any other
- f. The Tester checks that no Variable PDO has the same voltage range as any other
- g. The Tester checks that no Battery PDO has the same voltage range as any other.
- h. The Tester checks that no PPS/AVS PDO has the same voltage range as any other

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. Make sure to have run TEST.PREP.PR.1 preparation before this procedure. 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_{BUS} 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 (GiveBack Flag) = 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 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries and the test stops here. [COMMON.PROC.BU.1#4]
- 8. The check fails if *PS_RDY* Message is not received within *tPSTransition* min from the last bit of the EOP of *Accept* Message. [COMMON.PROC.BU.1#3]
- 9. An explicit contract is now established.
- 10. The Tester waits 500ms 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.
- 3. If the UUT attachment is detected, the Tester applies VCONN (if Ra 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. 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 100mA
- 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* (100 ms to 200 ms). 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. If the V_{BUS} voltage is stable within the target voltage, the Tester sends a *PS_RDY* Message at *tPSTransition* min after the reception of *Accept* Message.
10. An explicit contract is now established.
11. The Tester presents *SinkTxOk* if the test is in PD3 mode. The Tester waits 500ms 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 150ms nominal (*tCCDebounce*),
 - 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. Make sure to have run TEST.PREP.PR.1 preparation before this procedure. 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 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 4V 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) = 4V
 - 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 *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for 10 times or until *Accept* Message received. The check fails if the UUT responds with *Wait* Message after 10 retries and the test stops here. [COMMON.PROC.BU.4#4]
 8. The check fails if *PS_RDY* Message is not received within *tpSTransition* min from the last bit of the EOP of *Accept* Message. [COMMON.PROC.BU.4#3]
 9. An explicit contract is now established.
 10. The Tester waits 500ms 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*.
3. If the UUT attachment is detected, the Tester applies *VCONN* (if *Ra* is detected) and *vSafe5V* on *V_{BUS}*.
4. The Tester transmits *Source_Capabilities* Message with two PDOs:
 - a. 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. 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 5.9V
 - v. B15...8 (Minimum Voltage) set to 3.3V

- vi. B6...0 (Maximum Current) set to 3A
- 5. Repeat Step-4 if the Tester does not receive a *GoodCRC* from the UUT in response to *Source_Capabilities* Message. This requires a delay of *tTypeCSendSourceCap* (100 ms to 200 ms). 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 *tPSTransition* min after the reception of *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 *tPSTransition* min after the reception of *Accept* Message.
- 13. The Tester presents *SinkTxOK*. The Tester waits 500ms 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 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 VCONN 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 150ms ($tCCDebounce$),
 - 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} .
- 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 100mA
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* (100 ms to 200 ms). 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 *tpSTransition* min after the reception of *Accept* Message.
 12. The Tester waits 500ms 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. Make sure to have run TEST.PREP.PR.1 preparation before this procedure. 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 100mA
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* (100 ms to 200 ms). 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. If the V_{BUS} voltage is stable within the target voltage, the Tester sends a *PS_RDY* Message at *tPSTransition* min after the reception of *Accept* Message.
10. An explicit contract is now established.
11. The Tester presents *SinkTxOk* if the test is in PD3 mode. The Tester waits 500ms 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* + 15ms for 10 times. Once the Tester has resent *PR_Swap* for 10 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 *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 *Rp* 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 (GiveBack Flag) = 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 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* message after 10 retries, and the test stops here. [COMMON.PROC.BU.7#10]
23. The check fails if *PS_RDY* Message is not received within *tpSTransition* min from the last bit of the EOP of *Accept* Message. [COMMON.PROC.BU.7#8]
24. An explicit contract is now established.
25. The Tester waits 500ms 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_{BUS}* 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 (GiveBack Flag) = 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 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries, and the test stops here. [COMMON.PROC.BU.8#9]
7. The check fails if *PS_RDY* Message is not received within *tpSTransition* min from the last bit of the EOP of *Accept* Message. [COMMON.PROC.BU.8#3]

8. An explicit contract is now established.
9. The Tester waits 500ms 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* + 15ms for 10 times. Once the Tester has resent *PR_Swap* for 10 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_{BUS} 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 *vSafe5V* on V_{BUS} 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 @ 100mA, 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 500ms to respond to messages from the UUT.
18. 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.8#8]

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

1. Make sure to have run TEST.PREP.PR.1 preparation before this procedure. 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 100mA
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* (100 ms to 200 ms). 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. If the V_{BUS} voltage is stable within the target voltage, the Tester sends a *PS_RDY* Message at *tPSTransition* min after the reception of *Accept* Message.
 10. An explicit contract is now established.
 11. The Tester presents *SinkTxOk* if the test is in PD3 mode. The Tester waits 500ms 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* + 15ms for 10 times. Once the Tester has resent *PR_Swap* for 10 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 Rp 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 4V 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) = 4V
 - 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 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries, and the test stops here. [COMMON.PROC.BU.9#10]
24. The check fails if *PS_RDY* Message is not received within *tPSTransition* min from the last bit of the EOP of *Accept* Message. [COMMON.PROC.BU.9#8]
25. An explicit contract is now established.
26. The Tester waits 500ms 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 *Rp* to each of the CC1 and CC2 pins at the tester receptacle.
4. The Tester checks that it detects *Ra* on one pin (from now on called the VCONN pin) and *Rd* 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 *Rp* connected to the CC pin
 - b. waits 150ms (*tCCDebounce*),
 - c. disconnects *Rp* from the VCONN pin and applies VCONN (at the voltage specified in the Test Condition) to the VCONN pin,
6. The Tester waits for *tVCONNStable* max from when VCONN is stable.

NOTE:

1. VPD tests need to be run twice, with VCONN 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 Rd to each of the CC1 and CC2 pins at the captive cable end.
3. Tester waits for Rp to appear on exactly one pin. If this does not occur within *tDRP*, the check fails. [COMMON.PROC.BU.11#1].
4. The pin on which Rp appears from now on is called the CC pin. The other pin is referred to as the VCONN pin.
5. The Tester keeps Rd connected to the CC pin,
6. Check that V_{BUS} is detected at *vSafe5V* within *tCCDebounce* max. [COMMON.PROC.BU.11#2]
7. The Tester disconnects Rd from the VCONN pin.
8. Check that the first bit of a *Source_Capabilities* Message is received from the UUT within *tFirstSourceCap* max after VBUS present (*vSafe5V*). [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 (GiveBack Flag) = 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 10 times or until *Accept* Message is received. The check fails if the UUT responds with Wait Message after 10 retries, and the test stops here. [COMMON.PROC.BU.11#6]
12. The check fails if *PS_RDY* Message is not received within *tPSTransition* min from the last bit of the EOP of *Accept* Message. [COMMON.PROC.BU.11#5]

13. The Tester waits 500ms 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 VCONN 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 V_{BUS} present (*vSafe5V*). [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 (GiveBack Flag) = 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 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries, and the test stops here. [COMMON.PROC.BU.12#8]
7. The check fails if *PS_RDY* Message is not received within *tpSTransition* min from the last bit of the EOP of *Accept* Message. [COMMON.PROC.BU.12#3]
8. An explicit contract is now established.

9. The Tester waits 500ms 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* + 15ms for 10 times. Once the Tester has resent *PR_Swap* for 10 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_{BUS}* 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 presents *vSafe5V* on *V_{BUS}* 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 two PDOs:
 - a. 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. 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 5.9V
 - v. B15...8 (Minimum Voltage) set to 3.3V
 - 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 *tPSTransition* min after the reception of *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 *tPSTransition* min after the reception of *Accept* Message.
 24. The Tester presents *SinkTxOK*. The Tester waits 500ms 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 100mA
 - 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. If the V_{BUS} voltage is stable within the target voltage, the Tester sends a *PS_RDY* Message at *tPSTransition* min after the reception of *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, the Tester responds with a "Responder ACK" advertising Passive Cable with V_{BUS} Current Handling Capability set to 5A and Maximum V_{BUS} Voltage set to 20V.
 - 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. If the V_{BUS} voltage is stable within the target voltage, the Tester sends a *PS_RDY* Message at *tPSTransition* min after the reception of *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 (GiveBack Flag) = 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 *Accept* or *Wait* Message is not received, and the test stops here [COMMON.PROC.PD.11#2]

4. If the UUT responds with Wait Message, the Tester resends *Request* Message every *tSinkRequest* for 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries, and the test stops here. [COMMON.PROC.PD.11#4]
5. The check fails if *PS_RDY* Message is not received within *tpSTransition* min from the last bit of the EOP of *Accept* Message. [COMMON.PROC.PD.11#3]

COMMON.PROC.PD.12 UUT Sent DR_Swap

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

1. The Tester receives *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.17 Tester Sent VCONN_Swap Message

Procedure Applicability: The Tester runs this procedure wherever COMMON.PROC.PD.17 is referred 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 and the Tester resends *VCONN_Swap* every *tVCONNSwapWait* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *VCONN_Swap* for 10 times.
 2. 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
 3. The check fails if the UUT responds with an *Accept* Message and the VIF field *VCONN_Swap_To_On_Supported* = No
 4. The test ends here if the UUT does not send an *Accept* Message.

- 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 a *Wait* Message and the Tester resends *VCONN_Swap* every *tVCONNSwapWait* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *VCONN_Swap* for 10 times.
 - 2. 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_Off_Supported* = Yes
 - 3. The check fails if the UUT responds with an *Accept* Message and the VIF field *VCONN_Swap_To_Off_Supported* = No
 - 4. The test ends here if the UUT does not send an *Accept* Message.
 - 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]

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_Cap* message

Procedure Applicability: Unless otherwise specified, the Tester runs this procedure whenever receiving an *EPR_Source_Cap* message from the UUT.

1. The Tester receives the *EPR_Source_Cap* message from the UUT. This message is valid except 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_Cap* message with no EPR PDO, Object Position = 001b
 - ii. For *EPR_Source_Cap* message with EPR PDOs, Object Position = 1000b
 - b. B27 (GiveBack Flag) = 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 *PS_RDY* Message is not received within *tpSTransition* min (EPR mode) from the last bit of the EOP of *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 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 cap” bit is set to 0 in the Sourcecap message (SPR mode), Tester sends *Not_Supported* Message
 - b. The Tester is a Source only or Sink with Dual role power, and if the “EPR cap” bit is set to 1 in the Sourcecap message (EPR mode),
 - i. The Tester sends with *EPR_Source_Capabilities* Message that contains eight PDO’s and one PDO (140W 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. B23 set to 1b (*EPR_Mode* Capable)
 9. B22 (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: All fields set to 0

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

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

3. B27..B26(reserved) set to 00b

4. B25..B17(maximum voltage) set to 28V

5. B16(reserved) set to 0b

6. B15..B8 (minimum voltage) set to 15V

7. B7..B0 (PDP) set to 140

Tester checks below steps only in EPR mode:

ii. The check fails if the UUT does not respond with an *EPR_Request* Message.
[COMMON.PROC.PD3.3#2]

iii. The Tester responds with an *Accept* Message to the *EPR_Request* Message.

iv. If the V_{BUS} voltage is stable within the target voltage, the Tester sends a *PS_RDY* Message at *tPSTransition.EPRMode* min after the reception of 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 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. If the V_{BUS} voltage is stable within the target voltage, the Tester sends a *PS_RDY* Message at *tPSTransition.EPRMode* min after the reception of the *Accept* Message.

COMMON.PROC.PD3.5 Tester Sent *EPR_Mode* (Enter)

Procedure Applicability: The Tester runs this procedure wherever COMMON.PROC.PD3.5 is referred in the test procedure.

1. 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.
[COMMON.PROC.PD3.5#1]
2. Tester performs below checks,
 - a. If the VIF field Captive_Cable is No [AND] UUT is not a Vconn source, The Tester checks that UUT sends a *VCONN_Swap* message and completes the Vconn Swap sequence using COMMON.PROC.PD.6. [COMMON.PROC.PD3.5#2]
 - b. Else, the check fails if the UUT sends a *VCONN_Swap* message.
[COMMON.PROC.PD3.5#3]
3. Tester performs below checks,
 - a. If the VIF field Captive_Cable is No, the check fails if the UUT does not sends a SOP' *Discover Identity* Command. [COMMON.PROC.PD3.5#4]
 - b. Else, the check fails if the UUT sends a SOP' *Discover Identity* Command.
[COMMON.PROC.PD3.5#5]
4. 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* (500ms), 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.
[COMMON.PROC.PD3.5#6]
5. 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#7]
6. Tester checks that the UUT establishes EPR contract using common procedure
COMMON.PROC.PD3.2[COMMON.PROC.PD3.5#8]

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 Vconn 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 Vconn 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_{BUS} Current Handling Capability set to 10b(5A)
 - b. Maximum V_{BUS} 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. 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. Tester sends *EPR_Source_Capabilities* Message (as per the COMMON.PROC.PD3.3 step 2.b.i) with in 50ms from the last bit of the *GoodCRC* Message acknowledging the *EPR_Mode* (Enter_Succeeded) Message and tester checks that the UUT responds with *EPR_Request* Message. [COMMON.PROC.PD3.6#6]
8. 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 *tSinkEPRKeepAlive* 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 7 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

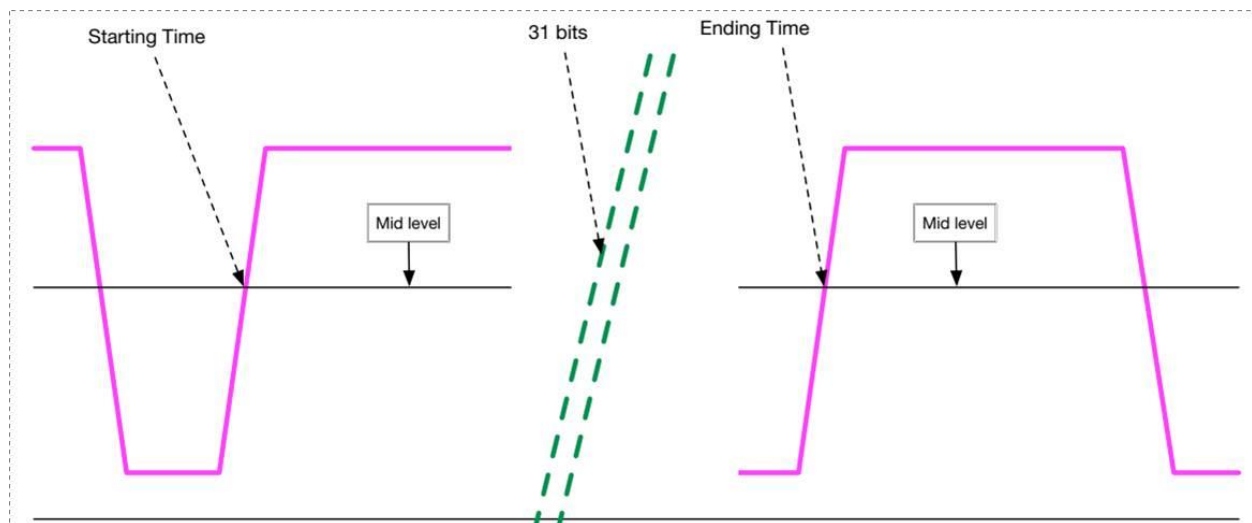


Figure 1: using mid-level to find the reference unit interval

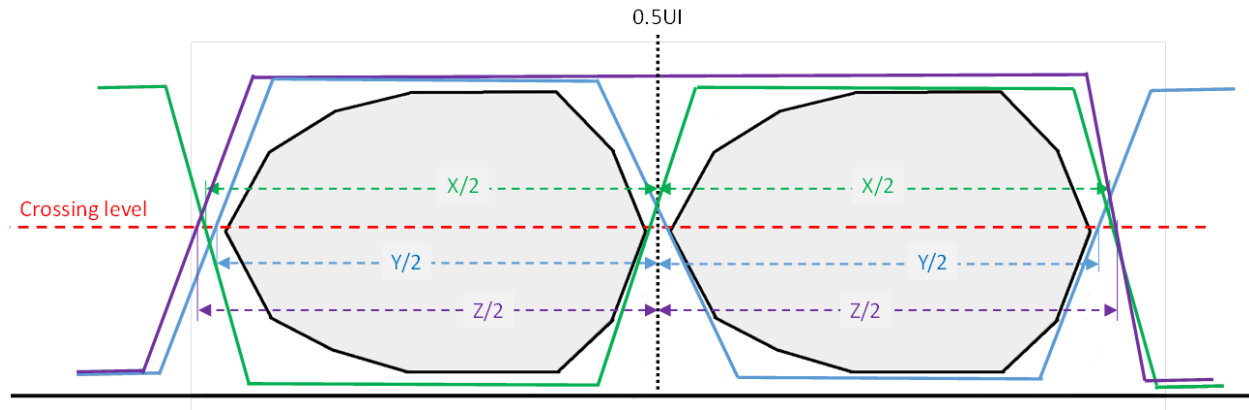


Figure 2: Example of producing eye diagram

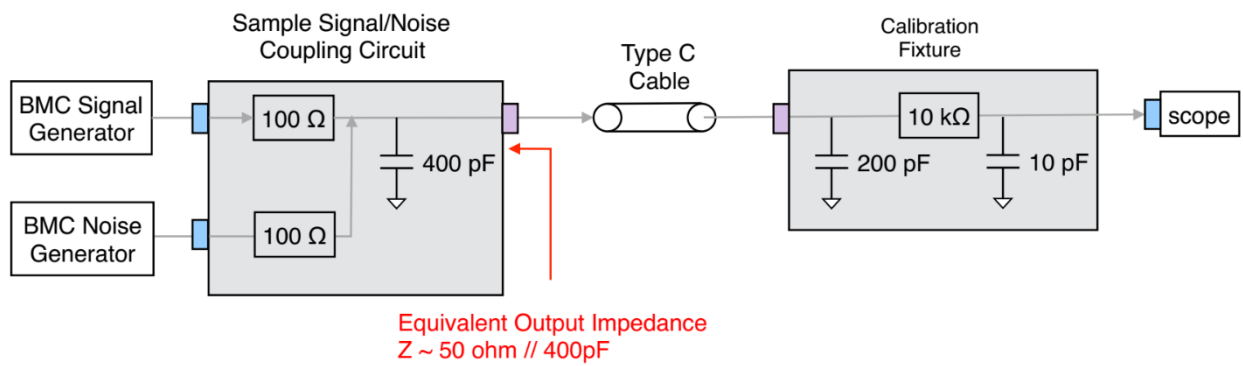


Figure 3: Noise injection generator and calibration fixture setup

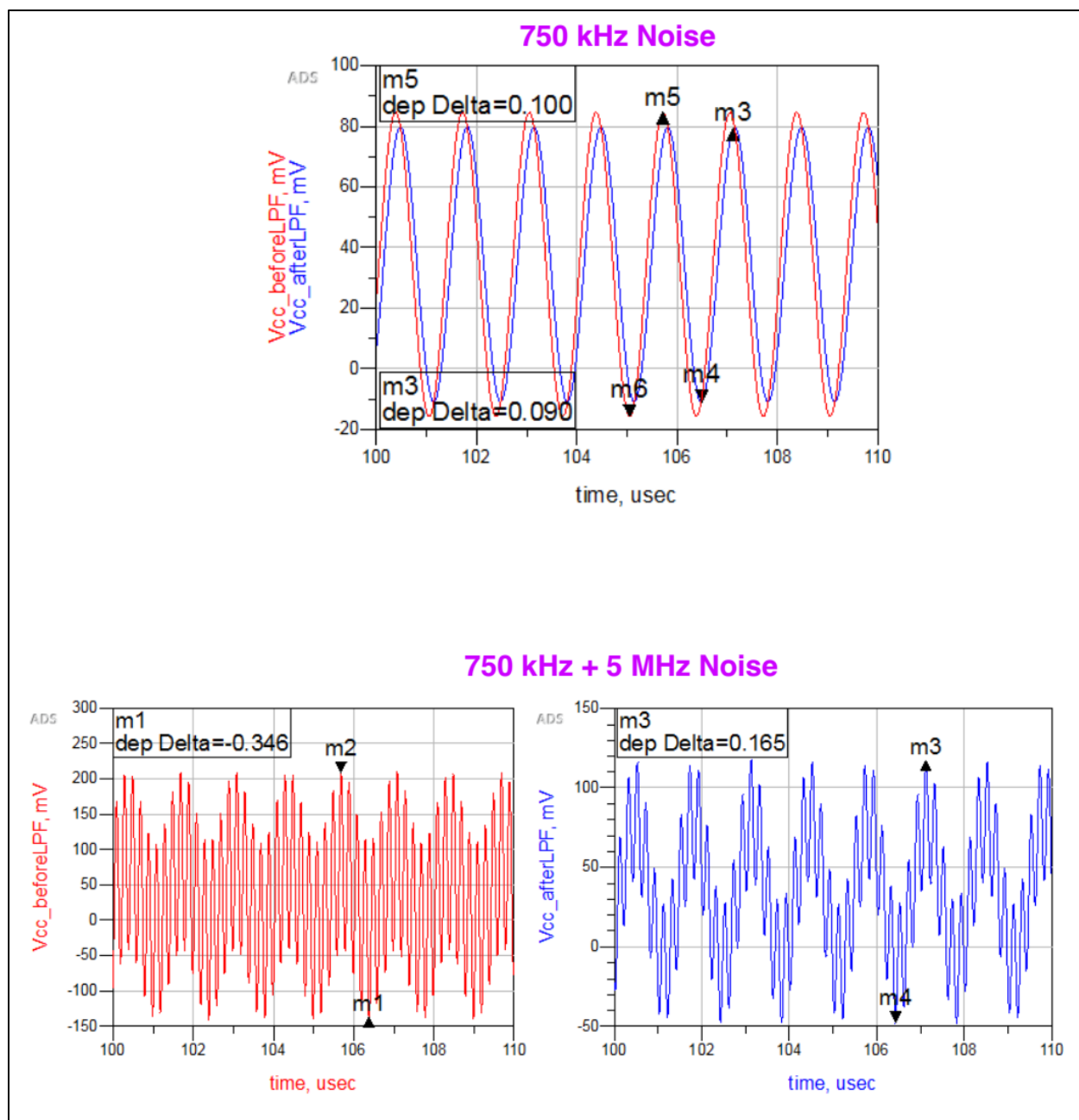


Figure 4: Two-tone noise calibration

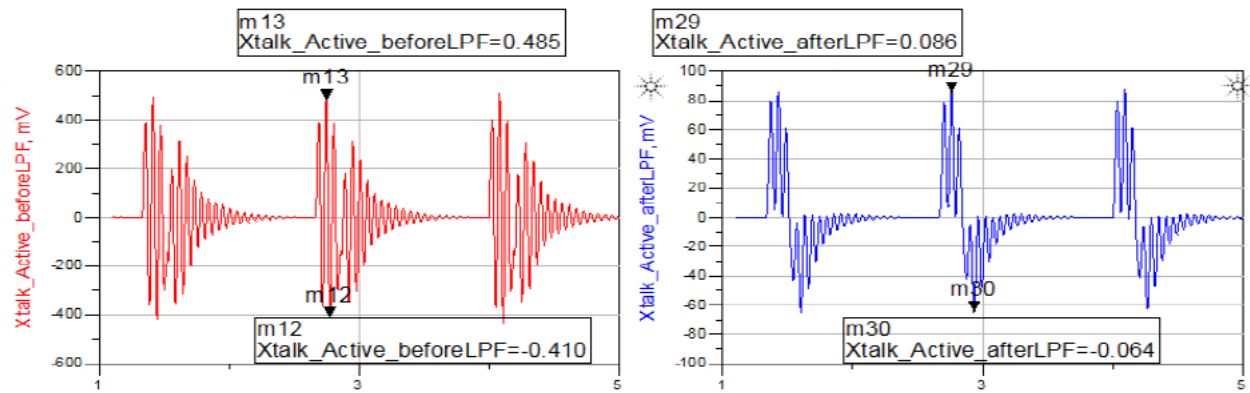


Figure 5: AWG measurement waveforms

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 Rp as a 4.7kΩ 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 Rd as a 5.1kΩ 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}, VCONN, Rp (4.7kΩ to 3.3V) and Rd (5.1kΩ) 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 Ω 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.
- Add 10mv to the initial crossing level to the mid-level, run Step-9
 - Subtract 10mv from the initial crossing level, run Step-9
 - Add 20mv to the initial crossing level, run Step-9
 - Subtract 20mv from the initial crossing level, run Step-9
 - Add 30mv to the initial crossing level, run Step-9
 - Subtract 30mv from the initial crossing level, run Step-9
 - Add 40mv to the initial crossing level, run Step-9
 - Subtract 40mv from the initial crossing level, run Step-9
 - Add 50mv to the initial crossing level, run Step-9
 - 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 “10%-Level” is defined as “low level” + (“high level” - “low level”) *10/100
 - A “90%-Level” is defined as “low level” + (“high level” - “low level”) *90/100
 - 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*.
 - 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
Rev3ChkdSrc		✓			
Rev3ChkdSnk	✓		✓		
Rev3eVcMinEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev3VcMinEnd2				✓	

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 15ms.
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 $t_{UnitInterval}$ max) to generate 3 transitions ($n_{TransitionCount}$). So that is 25us ($t_{InterFrameGap}$ minus 7.4us).*
6. The Tester waits for 10ms 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 10ms 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
Rev3ChkdSrc		✓			
Rev3ChkdSnk	✓		✓		
Rev3VcMinEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev3VcMinEnd2				✓	
Rev3VcMaxEnd2				✓	

Test Procedures:

- There are 6 possible bring-up procedures:
 - 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
 - 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.
 - 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
 - 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
 - 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.
 - 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.
- The Tester sends a BIST request to the UUT, specifying "BIST Test Data".
- The Tester waits 15ms.
- 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.

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 15ms.
4. Using TX_GROUP_1 parameters in the [Table 7](#) 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 7](#) 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 7](#) 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, the Tester performs the following checks [\[TEST.PD.PHY.ALL.5#4\]](#):
 - i. Using TX_GROUP_2 parameters in the [Table 7](#) 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.
 - ii. Using TX_GROUP_3 parameters in the [Table 7](#) 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.
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 ± 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. 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.

[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
Rev3ChkdSrc		✓			
Rev3ChkdSnk	✓		✓		
Rev3VcMinEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev3VcMinEnd2				✓	
Rev3VcMaxEnd2				✓	

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 15ms 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
Rev3ChkdSrc		✓			
Rev3ChkdSnk	✓		✓		
Rev3VcMinEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev3VcMinEnd2				✓	

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 Source or UUT is not actively sourcing Vconn)

and 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 = YES

the 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_Tyoe != eMarker **and**
 - iii. Captive_Cable_Is_eMarked = NO **and**
 - iv. (UUT is not the Vconn Source or UUT is not actively sourcing Vconn)
 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
Rev3ChkdSrc		✓			
Rev3ChkdSnk	✓		✓		
Rev3VcMinEnd1				✓	✓
Rev3VcMaxEnd1				✓	✓
Rev3VcMinEnd2				✓	
Rev3VcMaxEnd2				✓	

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 V_{BUS} 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:

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 BIST request to the UUT, specifying “BIST Test Data”.
3. The Tester waits 15ms to respond to messages from the UUT.
4. The Tester sends a *Cable Reset* Signal.

5. 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]
6. The Tester transmits *Hard Reset* flipping Bit 0 of the first two symbols (number 1&2).
7. The Tester waits 1.5 second and checks that the UUT does not send any message or removing V_{BUS} .
8. The Tester sends a *Get_Sink_Cap* Message to the UUT.
9. The Tester waits for *tCtsSenderResponse* (see [Table 16](#)), and the check fails if the UUT sends a response to *Get_Sink_Cap* Message. [TEST.PD.PHY.PORT.1#2]
10. 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.

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 @ 100mA, 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 tInterframeGap. 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 @ 100mA, 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*.
[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 *tInterframeGap* (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 @ 100mA, 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 500ms 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 500ms 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 send 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 11111b (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 *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
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. 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: Bit 0 and 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: values must be less than 7
 3. Bit5 (reserved) is 0.
 - 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: the check fails if
 - a. Manufacturer String in Manufacturer Info Data Block is not null terminated ASCII text string "Not Supported"
 - b. VID field is not 0xFFFF

[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: the check fails if:
 - a. Manufacturer String in Manufacturer Info Data Block is not null terminated ASCII text string "Not Supported"
 - b. VID field is not 0xFFFF

[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 5th 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 = 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 waits for tChunkSenderResponse max + 5 ms and then it sends the first chunk message to the UUT.
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 = 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

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	✓		✓		
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 *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:

5. 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 RMDO, B31...28 Revision.major matches to the VIF PD_Spec_Revision_Major field
- c. In RMDO, 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 (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.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:

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.PORT3.3#1]
 - a. If VIF field PD_Port_Type is not Consumer only, and the UUT does not send 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.PORT3.3#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
 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
 - 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 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 *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. Length = (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 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 Country_Info message. [TEST.PD.PROT.PORT3.6#2]
6. The Tester runs the following additional field checks for CIBD: [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:

- 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 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 The Tester sends an extended message to the UUT, with Data Size set to 260 and Message Type set to 11111b. Bytes 0 to 259 of data block contain incrementing values (mod 256) starting at 0x00.
- 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 (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.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 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.1#1]
4. 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:

- 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 sends a *Get_Source_Cap* Message to the UUT.
- The check fails if the Tester does not receive *Source_Capabilities* Message. [TEST.PD.PROT.SRC.2#1]
- The Tester does not send a *Request* Message after receiving *Source_Capabilities* Message.
- The Tester checks that a *Hard Reset* is detected between *tSenderResponse* min and max (see **Table 16**), the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the first bit of *Hard Reset* SOP 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_{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.SRC.3#1]
4. The Tester responds with a *Request* Message at *tCtsSnkRequest* (see Table 16) 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_{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.SRC.4#1]
4. The Tester sends a Request (starting from the first PDO), requesting 200% of the current or power offered.
5. The check fails if a *Reject* Message is not received. [TEST.PD.PROT.SRC.4#2]
6. 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 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.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_{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.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* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *DR_Swap* for 10 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* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *DR_Swap* for 10 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.
- 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 V_{BUS} Current Handling Capability set to 5A and Maximum V_{BUS} Voltage set to 50V.
3. 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* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *VCONN_Swap* for 10 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. 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* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *VCONN_Swap* for 10 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]

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 + 15ms for 10 times. Once the Tester has resent *PR_Swap* for 10 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 V_{BUS} 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 @ 100mA, 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* + 15ms for 10 times. Once the Tester has resent *PR_Swap* for 10 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 V_{BUS} voltage to *vSafe0V* within *tSrcSwapStdby* 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:

- 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 sends a *PR_Swap* Message.
- The Tester checks the response. [TEST.PD.PROT.SRC.10#1]
 - If the UUT responds with a *Wait* Message and the Tester resends *PR_Swap* every *tPRSwapWait* + 15ms for 10 times. Once the Tester has resent *PR_Swap* for 10 times, the check fails, and the test ends here.
 - 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.
 - 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.
- The Tester draws no more than iSwapStandby within *tSrcTransition* min after it has sent the last *GoodCRC*.
- The Tester checks the response: [TEST.PD.PROT.SRC.10#2]
 - The check fails if the UUT does not send a *PS_RDY* Message after the UUT driving V_{BUS} voltage to *vSafe0V*
 - 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*.
- 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.
- A USB Type-C Error Recovery is expected from the UUT. The Tester checks the response: [TEST.PD.PROT.SRC.10#3]
 - 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.
 - 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* + 15ms for 10 times. Once the Tester has resent *PR_Swap* for 10 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 15ms 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 (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.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* in order to force a SourceCapabilityTimer timeout, then verifies it is correctly implemented.

Test Specific Tester Behavior: N/A

Test Conditions:

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

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 201.1 ms (*tTypeCSendSourceCap* max + *tReceive* max)
 - ii. The check fails if this time is smaller than 100.9 ms (*tTypeCSendSourceCap* min + *tReceive* min)
 - 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
Rev3ChkdSrc		✓

Test Procedures:

1. The test starts in a disconnected state.
2. The Tester applies Rd, 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 detected between *tSenderResponse* min and max (see Table 16), the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the first bit of *Hard Reset* SOP 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
Rev3ChkdSrc		✓
Rev3UnchkdSrc		✓

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_Extended* Message to the UUT.
3. 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.
 - 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 an *Alert* Message to the UUT with Type of *Alert* field = Source Input Change Event (Bit 5).
- The test stops here when one of the following conditions happens:
 - A *Not_Supported* Message is received as a response to the last *Alert* Message.
 - A Get_Status message is not received within 500 ms after the last *Alert* Message was sent

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. Upon receiving the Get_Battery_Status message, the Tester checks Battery Status Ref = 0.
[TEST.PD.PROT.SRC3.5#1]
4. The test stops here when one of the following conditions happens:
 - a. A *Not_Supported* Message is received as a response to the last *Alert* Message.
 - b. A Get_Battery_Status message is not received within 500 ms after the last *Alert* Message was sent

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:

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. Upon receipt of the *Source_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 $t_{Receive} \text{ max} + t_{SoftReset} \text{ max}$, 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
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 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 Tester sends a Get_PPS_Status message to the UUT.
- The check fails if the Tester does not receive PPS_Status message. [TEST.PD.PROT.SRC3.7#1]
- The Tester runs the following additional field checks for the PPSSDB: [TEST.PD.PROT.SRC3.7#2]
 - Output Voltage field shall be within $\pm 3\%$ of the actual output voltage rounded to the nearest 20mV or 0xFFFF
 - Output Current field shall be 0x00 to 0x03(inclusive) or 0xFF
 - For Real Time Flags field
 - Bit 0 shall be 0
 - PTF shall be 0x00 (Not Supported) or 0x01 (Normal)
 - OMF shall be 0 (Constant Voltage mode)
 - 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
Rev3ChkdSrc		✓

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.SRC3.8#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.
- The Tester sends the second *Request* Message to request the Programmable Power Supply APDO at 4V 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]
- 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]
- If the UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries, and the test stops here. [TEST.PD.PROT.SRC3.8#6]
- The check fails if *PS_RDY* Message is not received within *tPSTransition* min from the last bit of the EOP of *Accept* Message, and the test stops here. [TEST.PD.PROT.SRC3.8#3]
- 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

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 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 4V 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
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 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.SRC3.10#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.

3. 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]
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 the check with a warning, and the test stops here. [TEST.PD.PROT.SRC3.10#3]
5. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries, and the test stops here. [TEST.PD.PROT.SRC3.10#5]
6. The check fails if *PS_RDY* Message is not received within *tPSTransition* min from the last bit of the EOP of *Accept* Message, and the test stops here. [TEST.PD.PROT.SRC3.10#6]
7. The check fails if a *Hard Reset* is received within *tPPSTimeout* max + 15ms. [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_{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.SRC3.11#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:

- 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 sends a *FR_Swap* Message to the UUT.
- The Tester checks the response: [TEST.PD.PROT.SRC3.12#1]
 - 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*.
 - 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
<i>Rev3ChkdSrc</i>		✓

Test Procedures:

1. The test starts in a disconnected state.
2. The Tester applies R_d and R_a , and it waits for V_{BUS} present.
3. 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_{BUS} Current Handling Capability set to 5A and Maximum V_{BUS} 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]
4. The Tester removes all terminations and simulates a disconnection
5. The Tester applies R_d and R_a , and it waits for V_{BUS} present.
6. The Tester checks the response:
 - a. If a SOP' *Discover Identity* Command is received, the Tester replies as a Passive Cable with V_{BUS} Current Handling Capability set to 3A and Maximum V_{BUS} 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]
7. The Tester removes all terminations and simulates a disconnection
8. The Tester applies R_d but does not present R_a , and it waits for V_{BUS} present.
9. The Tester shall disregard *Discover Identity* SOP' messages if any.
10. 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: The test is applicable if one of the following is true:

- Port_Source_power_Type VIF field = 1 (Shared) **or**
- Invariant_PDOs_Port VIF field - YES

	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, and the check fails if one of the following conditions is met:
[TEST.PD.PROT.SRC3.14#1]
 - a The UUT replies with *Not_Supported* Message
 - b The Number of Data Objects in the Source_Info message is not 1
 - c In SIDO, B31 (Port Type) does not match to the Port_Managed_Guaranteed_Type VIF field
 - d In SIDO, B30...24 are not Zeros
 - e In SIDO, B23...16 (Port Maximum PDP) is not equal to the PD_Power_As_Source VIF field
 - f In SIDO, B15...8 (Port Present PDP) is either zero or higher than Maximum PDP
 - g In SIDO, B7...0 (Port Reported PDP) is either zero or higher than Maximum PDP
 - h If the Port_Managed_Guaranteed_Type VIF field is set to 1 (Guaranteed), and B23...16 (Port Maximum PDP) is not equal to B15...B8 (Port Present PDP).

5.4 Test Procedures for Sink Capable UUT

5.4.1 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 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.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:

- 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_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.
- 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_{BUS} to *vSafe0V* and then restores V_{BUS} to *vSafe5V*.
- The Tester transmits *Source_Capabilities* Message with single PDO immediately prior to *tTypeCSinkWaitCap* min (the delay is from the time V_{BUS} 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
 - B21...20 (Peak Current) set to 00b
 - B19...10 (Voltage) set to 5V
 - B9...0 (Maximum Current) set to 100mA
- The Tester continues to present *SinkTxNG* while waiting for a response if the test is running in PD3 mode. The Tester checks that the UUT responds with a *Request* Message.

[TEST.PD.PROT.SNK.3#1]

TEST.PD.PROT.SNK.4 SinkWaitCapTimer Timeout

Description: The Tester does not send *Source_Capabilities* Message after cycling V_{BUS} 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:

- 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_{BUS} to *vSafe0V* and then restores V_{BUS} to *vSafe5V*.
- The Tester does not send a *Source_Capabilities* Message after cycling the V_{BUS} to force a *SinkWaitCapTimer* timeout on the UUT.
- The check fails 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_{BUS} 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:

- 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

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 100mA
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 16](#)) 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 first bit of the preamble of the *Accept* Message has been transmitted.
5. The check fails if a *Hard Reset* is detected within *tCtsSenderResponse* (see [Table 16](#)) 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 100mA
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 detected between *tSenderResponse* min and max (see [Table 16](#)), the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the first bit of *Hard Reset* SOP 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 @ 100mA, 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 *Accept* Message EOP and the first bit of Preamble 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:

- 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_{BUS} to *vSafe0V* and then restores V_{BUS} to *vSafe5V*.
- The Tester transmits *Source_Capabilities* Message with single PDO immediately prior to *tTypeCSinkWaitCap* min
 - 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
 - B21...20 (Peak Current) set to 00b
 - B19...10 (Voltage) set to 5V
 - B9...0 (Maximum Current) set to 100mA
- 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.
[TEST.PD.PROT.SNK.8#1]
- 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.
- The check fails if a *Soft Reset* Message is not received within *tProtErrSoftReset* max.
[TEST.PD.PROT.SNK.8#2]
- 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_{BUS} to *vSafeOV* and then restores V_{BUS} to *vSafe5V*. The Tester transmits *Source_Capabilities* Message with single PDO (5V @ 100mA, Source only) immediately prior to *tTypeCSinkWaitCap* min.
4. 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.9#1]
5. The Tester responds with an *Accept* Message to the *Request* Message.
6. The Tester sends a *Get_Sink_Cap* Message at *tpSTransition* min after the last bit of the EOP of the *Accept* Message sent.
7. The check fails if a *Hard Reset* is not received within 15ms 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:

- 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 requests a *DR_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.10#1]
 - If the UUT responds with a *Wait* Message, the Tester resends *DR_Swap* every *tDRSwapWait* + 15ms for 10 times. The check fails if the Tester has resent *DR_Swap* for 10 times and the test ends here.
 - The check fails if
 - In PD2, the UUT responds with a *Reject* Message and the VIF field DR_Swap_To_DFP_Supported = Yes.
 - In PD3, the UUT responds with either
 - Not_Supported* and (the VIF field DR_Swap_To_DFP_Supported = Yes or the VIF field DR_Swap_To_UFP_Supported = Yes) or
 - Reject* and the VIF field DR_Swap_To_DFP_Supported = Yes

If the check fails, the test ends here.

 - The check fails if the UUT responds with an *Accept* Message and the VIF field DR_Swap_To_DFP_Supported = No.
- If VIF field DR_Swap_To_DFP_Supportd=N, the test ends here. Otherwise, the Tester requests a further *DR_Swap*.
- The Tester checks the response. [TEST.PD.PROT.SNK.10#2]
 - If the UUT responds with a *Wait* Message, the Tester resends *DR_Swap* every *tDRSwapWait* + 15ms for 10 times. The check fails if the Tester has resent *DR_Swap* for 10 times.
 - The check fails if
 - In PD2, the UUT responds with a *Reject* Message and the VIF field DR_Swap_To_UFP_Supported = Yes.
 - 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: 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 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]
3. 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.
4. The Tester checks the response. [TEST.PD.PROT.SNK.11#2]
 - a. If the UUT responds with a *Wait* Message and the Tester resends *VCONN_Swap* every *tVCONNSwapWait* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *VCONN_Swap* for 10 times.
 - b. 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
 - c. The check fails if the UUT responds with an *Accept* Message and the VIF field VCONN_Swap_To_On_Supported = No
 - d. The test ends here if the UUT does not send an *Accept* Message.
5. The Tester checks that the UUT sends a *PS_RDY* Message. [TEST.PD.PROT.SNK.11#3]
6. The Tester checks that VCONN is present at the non-CC line. [TEST.PD.PROT.SNK.11#4]
7. The Tester requests a further *VCONN_Swap*.
8. The Tester checks the response. [TEST.PD.PROT.SNK.11#5]

- a. If the UUT's response is a Wait, the Tester waits *tVCONNSwapWait* + 15ms seconds to resend *VCONN_Swap*. The Tester sends *VCONN_Swap* up to 10 times, and the test ends here if the UUT does not respond with an *Accept* or *Reject* Message.
 - i. In PD3 mode and the UUT responds with a *Reject* Message, the check fails.
 - ii. 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* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *PR_Swap* for 10 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_{BUS} voltage to *vSafe0V* within *tSrcSwapStdby* max.
5. The Tester continues to assert Rp (i.e., the Tester does not assert Rd 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 $t_{PRSwapWait} + 15\text{ms}$ for 10 times. The check fails and the test ends here if the Tester has resent *PR_Swap* for 10 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 $t_{SrcTransition}$ max (this delay is from the last bit of *GoodCRC* sent in response to the *Accept* Message), the Tester drives the V_{BUS} voltage to *vSafe0V* within $t_{SrcSwapStdby}$ max.
 5. The Tester presents *Rd* on the CC wire. The Tester sends a *PS_RDY* at the deadline limit of $t_{PSSourceOff}$ 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 $t_{PSSourceOn}$ 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 $t_{SwapSourceStart}$ 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 500ms 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 detected between $t_{SenderResponse}$ min and max (see **Table 16**), the delay is between the last bit of the *GoodCRC* Message EOP has been sent and the first bit of *Hard Reset* SOP 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.
3. If the UUT attachment is detected, the Tester applies VCONN (if Ra is detected) and *vSafe5V* on V_{BUS}.
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 100mA
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* (100 ms to 200 ms). 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. If the V_{BUS} voltage is stable within the target voltage, the Tester sends a *PS_RDY* Message at *tPSTransition* min after the reception of *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:

- 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 with Type of *Alert* field = OTP event (Bit 3).
- The Tester presents *SinkTxOK*.
- The test stops here when one of the following conditions happens:
 - A *Not_Supported* Message is received as a response to the last *Alert* Message
 - A Get_Status message is received as a response to the last *Alert* Message
 - No Get_Status message is received within 500 ms after the last *Alert* Message was sent

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:

- 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 = Battery Status Change Event (Bit 1)
 - b. Fixed Batteries = Battery 0 (B20)
3. The Tester presents *SinkTxOK*.
4. The test stops here when one of the following conditions happens:
 - a. A *Not_Supported* Message is received as a response to the last *Alert* Message
 - b. A Get_Battery_Status message is received; the Tester checks that Battery Status Ref = 0. [TEST.PD.PROT.SNK3.3#1]
 - c. No Get_Battery_Status message is not received within 500 ms after the last *Alert* Message was sent.

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 *tReceive* max + *tSoftReset* 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:

- 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.12
 - 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
- The Tester presents SinkTxOk.
- The check fails if a *Request* Message is not received within *tPPSRequest* 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.
- The Tester runs the following additional field checks on the Request Data Object:
[TEST.PD.PROT.SNK3.5#2]
 - Object Position field is 2.

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:

- The test starts in a disconnected state.
- The tester applies Rp.

3. If the UUT attachment is detected, the Tester applies VCONN (if Ra is detected) and *vSafe5V* on V_{BUS} .
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 100mA
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* (100 ms to 200 ms). This sequence is repeated at least nCapsCount times.
7. 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:

1. The test starts in a disconnected state.
2. The tester applies Rp.
3. If the UUT attachment is detected, the Tester applies VCONN (if Ra 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 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 100mA
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* (100 ms to 200 ms). This sequence is repeated at least *nCapsCount* times.
7. The check fails if the UUT does not respond with a *Request* Message. [TEST.PD.PROT.SNK3.7#1]
8. 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]
9. Repeat the test with Specification Revision in *GoodCRC* set to 01b and 10b.

TEST.PD.PROT.SNK.8 GotoMin Message

Description: As a Source, the Tester verifies that the UUT behaves properly when it receives *GotoMin* Message.

Test Specific Tester Behavior: N/A

Test Conditions:

- The test is applicable if VIF field GiveBack_May_Be_Set is set to Yes
- The Power Contract is established where the Tester accepted to provide power with current equal to Operating Current.

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

Test Procedures:

1. There are 2 possible bring-up procedures:
 - i. 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

- ii. 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 pre-checks conditions: [TEST.PD.PROT.SNK.8#1]
 - i. The Request Data Object has the GiveBack flag (B27) set
 - ii. The Operating Current (B19..10) is higher than Minimum Operating Current (B9..0)
If any of the checks is not true, the Tester passes the test, and the test stops here.
3. The Tester sends the *GotoMin* Message to the UUT.
4. The Tester checks: [TEST.PD.PROT.SNK.8#2]
 - i. The UUT replies with the *GoodCRC* Message

The UUT transition to the new power consumption level with the current equal to the Minimum Operating Power within *tSrcTransition* (25mS)

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 follow: [TEST.PD.VDM.SRC.1#1]
 - i. The check fails If Supports Structured VDM is NO and the UUT responds with Responder NAK
 - ii. The check fails If Supports Structured VDM is YES and the UUT Ignores the *Discover Identity* Command and
 - iii. The check fails If the UUT responds with Responder ACK or Responder BUSY.
 - b. In PD3 mode, the Tester checks the response as follow: [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 *tVDMBusy* min after it is in PE_SRC_RDY for 10 times.
 - ii. The check fails if the UUT responds with a “Responder BUSY” message, and the Tester has resent *Discover Identity* Command for 10 times every *tVDMBusy* min
 - iii. The check fails If Supports Structured VDMs is NO and the UUT responds with Responder ACK
 - iv. The check fails if Responds_To_Discov_SOP_DFP is NO and the UUT responds with Responder ACK
 - v. The check fails If Responds_To_Discov_SOP_DFP is YES and the UUT responds with Responder NAK
 - vi. The check fails If Supports Structured VDMs is NO and the UUT responds with Responder NAK.
 - vii. The check fails If Supports Structured VDMs is YES and the UUT responds with Not Supported.
 - 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 16](#)), and the check fails if the UUT sends an
Attention Response message (ACK, NAK or BUSY). [\[TEST.PD.VDM.SRC.1#4\]](#)

[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:

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 Tester checks the response. [TEST.PD.VDM.SNK.1#1]
 - a. The UUT shall respond with a “Responder NAK, if VIF fields Responds_To_Discov_SOP_UFP = NO and and Supports Structured VDMs is Yes. Test ends here.
 - b. 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.
 - 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 10 times. The check fails and the test ends here if the Tester has resent *Discover Identity* Command for 10 times.
- d. The check fails if the UUT responds with a “Responder BUSY” message and the Tester has resent *Discover Identity* Command for 10 times every *tVDMBusy* min. Test ends here.
 - e. The check fails if the UUT responds with a “Responder ACK” message and VIF field Responds_To_Discov_SOP_UFP= No. Test ends here.
 - f. 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.
3. Perform the following check on “Responder ACK” message: [TEST.PD.VDM.SNK.1#2]
 - a. Number of VDOs:
 - i. The check fails if the number of VDOs in the message does not match the value stated in the Tables 4-1, 4-2, 4-3, 4-4 below, as appropriate, based on the Product Type(s) found in the message.
 - ii. The check fails if the number of VDOs is 7, and the 6th VDO (Pad) is not all 0’s.
 - b. ID Header VDO:
 - i. 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:
 - 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 10 times. The check fails and the test ends here if the Tester has resent *Discover Identity* Command for 10 times.
 - c. If the UUT responds a "Responder BUSY" message, the Tester resends *Discover SVIDs* Command for 10 times every *tVDMBusy* min. Once the Tester has repeated *Discover SVIDs* for 10 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 list SVID<X>_SOP. 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 SVID<X>_SOP in VIF.
 - ii. If SVID_Fixed_SOP is set to NO, each SVID in the VDO list can be in any position in the VIF's SVID<X>_SOP list.
 - 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 10 times. The check fails and the test ends here if the Tester has resent *Discover Identity* Command for 10 times.
 - iii. If the UUT responds a "Responder BUSY" message, the Tester resends *Discover Modes* Command for 10 times every *tVDMBusy* min. Once the Tester has repeated *Discover SVIDs* for 10 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<X>_Num_Modes_Min_SOP and SVID<X>_Num_Modes_Max_SOP.
- vii. Check that each Mode discovered matches one of the VIF list SVID<X>_Mode<Y>_SOP. The procedure for matching them is as follows:
 - 1. If SVID<X>_Modes_Fixed_SOP is set to YES, the match is done by simply comparing the value in the VDO with the corresponding value in the SVID<X>_Mode<Y>_Recog_Value_SOP. Both lists should be in the same sequence.
 - 2. If SVID<X>_Modes_Fixed_SOP is set to NO, the value in the VDO is bitwise ANDed with each SVID<X>_mode<Y>_Recog_Mask_SOP and the result is compared with the corresponding SVID<X>_Mode<Y>_Recog_Value_SOP 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.
 - b. The Tester checks the response. [TEST.PD.VDM.SNK.1#6]
 - i. If the UUT responds with “Responder BUSY”, the check fails.
 - ii. The check fails if the UUT responds with “Responder NAK” to the corresponding SVID<X>_mode<Y>_enter_SOP = Yes.
 - 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 16), and the check fails if the UUT sends an *Attention* Response message (ACK, NAK or BUSY). [TEST.PD.VDM.SNK.1#7]

Table 8 Rev 2 Port - Number of VDOs Expected

Product Type (UFP)	Undefined	4
	PD USB Hub	4
	PD USB Peripheral	4
	AMA	5

Table 9 Rev 2 Cable - Number of VDOs Expected

Product Type	PassiveCable	5
	Active Cable	5

Table 10 Rev 3 Port Number of VDOs Expected

Product Type (UFP)	Product Type (DFP)				
		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 6th VDO (Pad) must be all 0's

Table 11 Cable - Number of VDOs Expected

Product Type	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]
 - 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.
 - 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 10 times. The check fails and the test ends here if the Tester has resent *Discover Identity* Command for 10 times.
 - If the UUT responds with a BUSY message, the Tester resends *Discover SVIDs* Command for 10 times every *tVDMBusy* min. Once the Tester has repeated *Discover SVIDs* Command for 10 times, the check fails, and the test ends here.
 - If the UUT responds with a "Responder NAK" message, the Tester checks that VIF field Modal_Operation_Supported_SOP = No. The test ends here.
 - If the UUT responds with a "Responder ACK" message, the Tester checks that VIF field Modal_Operation_Supported_SOP = Yes.
- 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

Description: The Tester verifies that the UUT correctly supports PD message interruption during VDM exchanges.

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.
3. The Tester sends a *Get_Sink_Cap* using SOP packet immediately after *tInterFrameGap* has expired, the delay is from the last bit of the EOP of the UUT sent *GoodCRC*.
4. The check fails if the UUT does not send a *Sink_Capabilities* Message. [TEST.PD.VDM.SNK.3#1]
5. The Tester sends a *Discover SVIDs* Command to the UUT, using SOP.
6. The Tester sends a *Get_Sink_Cap* using SOP packet immediately after *tInterFrameGap* has expired, the delay is from the last bit of the EOP of the UUT sent *GoodCRC*.
7. The check fails if the UUT does not send a *Sink_Capabilities* Message. [TEST.PD.VDM.SNK.3#2]

TEST.PD.VDM.SNK.4 Interruption by VDM Message

Description: The Tester verifies that the UUT correctly aborts an ongoing VDM exchange if it is interrupted by other VDM command.

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 sends a *Discover SVIDs* Command using SOP packet immediately after tInterFrameGap has expired, the delay is from the last bit of the EOP of the UUT sent *GoodCRC*.
- The Tester checks the response: *[TEST.PD.VDM.SNK.4#1]*
 - 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.
 - If the UUT interrupts the Discovery Process (by responding with anything but a *Discover SVIDs* Response message), the Tester responds with appropriate response message and then resends *Discover SVIDs* Command *tVDMBusy* min after it is in PE_SNK_RDY for 10 times. The check fails and the test ends here if the Tester has resent *Discover SVIDs* Command for 10 times.
 - If the UUT does not respond with a “*Discover SVIDs* Responder ACK/NAK/BUSY”, the check fails.
 - If the UUT responds with a “*Discover Identity* Responder ACK/NAK/BUSY” message, the check fails.

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:

- 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 10 times. The check fails and the test ends here if the Tester has resent *Discover Identity* Command for 10 times.
 - c. If the UUT responds with a BUSY message, the Tester resends *Discover SVIDs* Command for 10 times every *tVDMBusy* min. Once the Tester has repeated *Discover SVIDs* Command for 10 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 10 times every *tVDMBusy* min. Once the Tester has repeated *Discover Modes* Command for 10 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 Mode<Y> in the SVID<X> where SVID<X>_mode<Y>_enter = Yes in the VIF file:
 - a. The Tester sends *Enter Mode* to the UUT, using SOP.
 - 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
Rev2Snk	✓	
Rev3ChkdSnk	✓	

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 Structured VDM Version field in the VDM Header is set to 3.
- The Tester checks the response **[TEST.PD.VDM.SNK.6#1]**
 - 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.
 - Otherwise (either VIF fields Responds_To_Discov_SOP_UFP = Yes or Responds_To_Discov_SOP_DFP = Yes), 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
Rev2Snk	✓	
Rev3ChkdSnk	✓	

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

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 (Vconn 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 10 times every *tVDMBusy* min. Once the Tester has repeated *Discover Identity* for 10 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 Reserved 0b
 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]
 - e. The check fails if the UUT responds anything but a *Discover SVIDs* Response message.
 - f. If the UUT responds with "Responder BUSY" message, the Tester resends *Discover SVIDs* Command for 10 times every *tVDMBusy* min. Once the Tester has repeated *Discover SVIDs* for 10 times, the check fails, and the test ends here.
 - g. If the UUT responds with a "Responder NAK" message, the check fails if VIF field Modal_Operation_Supported = YES. The test ends here.
 - h. 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]
 - i. Each discovered SVID appears in VIF list SVID<X>. 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 SVID<X> in VIF.
 - ii. If SVID_Fixed is set to NO, each SVID in the VDO list can be in any position in the VIF's SVID<X> list.
 - j. The number of SVIDs discovered lies between Num_SVIDs_min and Num_SVIDs_max.
 - k. For each VDO, the Tester checks if B15...0 (SVID n+1) are set to zeros, this marks the last VDO.
 - l. 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.
 - m. 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:
 - n. The Tester sends a *Discover Modes* Command to the UUT, using SOP'.
 - o. 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 10 times every *tVDMBusy* min. Once the Tester has repeated *Discover Modes* for 10 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<X>_Num_Modes_Min and SVID<X>_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 one of the VIF list SVID<X>_Mode<Y>. The procedure for matching them is as follows:
 - 1. If SVID<X>_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<X>_Mode<Y>_Recog_Value. Both lists should be in the same sequence.
 - 2. If SVID<X>_Modes_Fixed is set to NO, the value in the VDO is bitwise ANDed with each SVID<X>_mode<Y>_Recog_Mask and the result is compared with the corresponding SVID<X>_Mode<Y>_Recog_Value in turn until a match is found.
- 7. For every Mode in each SVID:
 - p. The Tester sends *Enter Mode* to the UUT, using SOP'.
 - q. The Tester checks the response: [TEST.PD.VDM.CBL.1#5]
 - i. If the UUT responds with "Responder BUSY", the check fails.
 - ii. The check fails if the UUT responds with "Responder NAK" to the corresponding SVID<X>_mode<Y>_enter = Yes. If the UUT responds with "Responder ACK" to *Enter Mode*, the Tester then sends an *Exit Mode* Command to the UUT. The check fails if the UUT does not respond with "Responder ACK" to *Exit Mode*.
- 8. The Tester sends *Attention* to the UUT. 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
Rev3VcMinEnd1				✓	✓

Rev3VcMaxEnd1				✓	✓
Rev3VcMinEnd2				✓	
Rev3VcMaxEnd2				✓	

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 *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 10 times every tVDMBusy min. Once the Tester has repeated *Discover Identity* for 10 times, the check fails if the UUT does not respond with a "Responder ACK" message.

6 Power Supply Specific Tests

Table 12 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 13 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 or *vPpsNew* min limit shall be (*vSrcNew* 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.BU.PD.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 μ 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 (GiveBack Flag) = 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 Request: As defined in Step 2.f.i
 2. B9...0 (Maximum Operating Current / Power) = Maximum Current / Power in the Source PDO.
 - vi. For PPS:

1. B19...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

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 10 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 *tPSTransition* min from the last bit of the EOP of *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 10 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 *tPSTransition* min from the last bit of the EOP of *Accept* Message, the check fails, and the test stops here.
[TEST.PD.PS.SRC.1#11]
 - v. The Tester responds to any *Get_Sink_Capabilities* 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 14).
 - vi. If Source Capabilities are received with less than the Sink Capabilities (same as VIF Source Capabilities) and 2 seconds have not elapsed, then the Tester goes back and repeats the *Request* Message in step I with the Mismatch bit set.
 - vii. 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.1#12]
 - viii. When the required Source Capabilities have been received by the Tester, it sends *Request* Message defined above.
 - ix. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of 10 times or until *Accept* is received.

- x. If *Accept* Message is not received, the check fails, and the test stops here.
 - xi. If a *PS_RDY* Message is not received within *tPSTransition* min from the last bit of the EOP of *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 μ 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 μ 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.SRC.1#2]
 - iii. If the transition involves a change of PDO, decrease the power drawn by the Tester to less than *pSnkStdby* within *tSnkStdby* of the last bit 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 *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*. [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* 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* or *vPpsNew*. [TEST.PD.PS.SRC.1#7]

6. After having sent *GoodCRC* in response to *PS_RDY*, the Tester increases its current draw to:

- a. For Fixed, Battery and Variable 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* 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):
- i. For Fixed, Battery and Variable PDOs, replace B19...10 (Operating Current / Power):
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 PDO, replace B19..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)

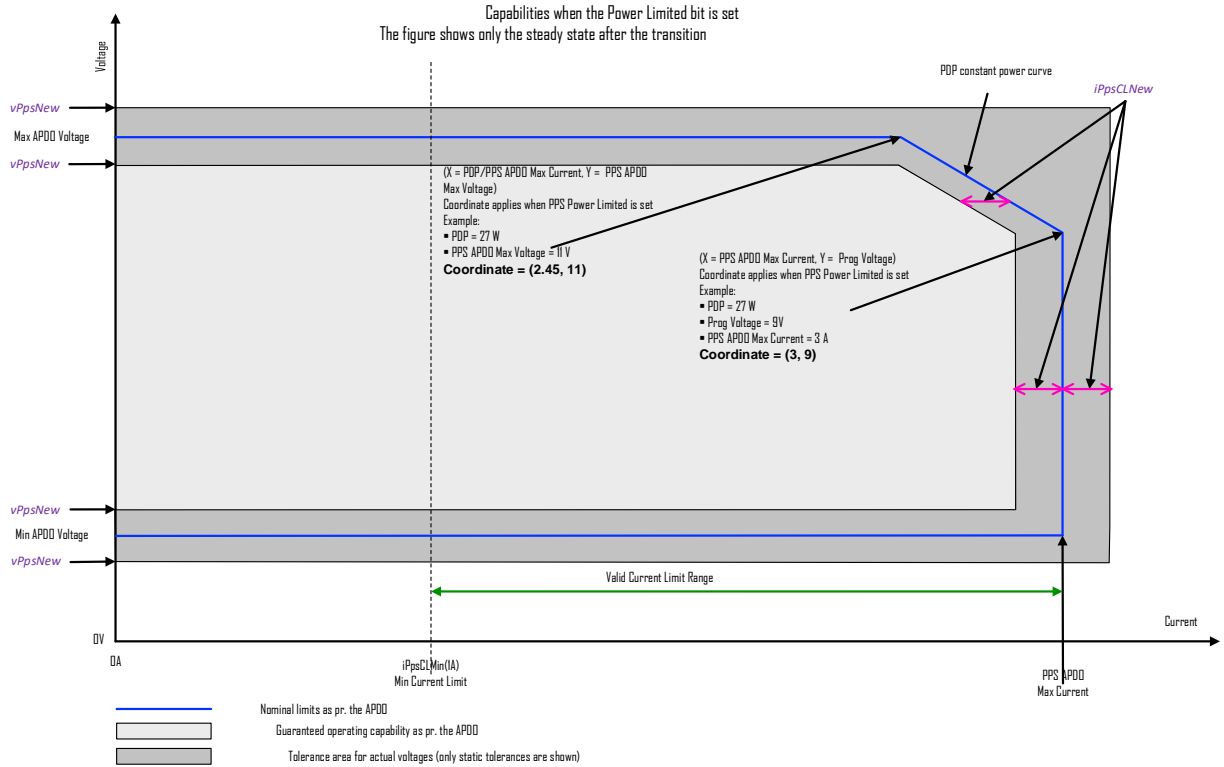


Figure 6 Voltage and Current Tolerances

Table 14 Sink Capabilities Extended Message

Field	Description
VID	Vednor ID of Tester
PID	Product ID of Tester
XID	0
FW Version	At Tester Vendor's Discretion
HW Version	At Tester Vendor's Discretion
SKEDB Version	1
Load Step	0
Sink Load Characteristics	0
Compliance	0
Touch Temp	0
Battery Info	0
Sink Modes	Bit 0: 1 if Source Caps in VIF include PPS, else 0 Bit 1: 1

	Bit 2: 0 Bit 3: 0 Bit 4: 0 Bit 5: 0 Bit 6: 0 Bit 7: 0
Sink Minimum PDP	Copy from VIF PD_Power_As_Source, maximum of 100
Sink Operational PDP	Copy from VIF PD_Power_As_Source, maximum of 100
Sink Maximum PDP	Copy from VIF PD_Power_As_Source, maximum of 100
EPR Sink Minimum PDP	0
EPR Sink Operational PDP	0
EPR Sink Maximum PDP	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.BU.PD.1.
2. The Tester repeats the following steps using the PDO as provided in the PDO Sequence Table (see [Table 12](#)), based on the Source PDOs specified in the VIF:
 - a. The Tester defines *Request* Message as following:
 - i. B27 (GiveBack Flag) = 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 PDO:

B19...9 (Output Voltage) = Maximum Voltage offered in Source APDO

B7...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 10 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 *tPSTransition* min from the last bit of the EOP of *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 10 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 *tPSTransition* min from the last bit of the EOP of *Accept* Message, the check fails, and the test stops here.
[TEST.PD.PS.SRC.2#2]
 - v. The Tester responds to any *Get_Sink_Capabilities* 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 14).
 - vi. If Source Capabilities are received with less than the Sink Capabilities (same as VIF Source Capabilities) and 2 seconds have not elapsed, then the Tester goes back and repeats the *Request* Message in step I with the Mismatch bit set.
 - vii. 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.2#8]
 - viii. When the required Source Capabilities have been received by the Tester, it sends *Request* Message defined above.
 - ix. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of 10 times or until *Accept* is received.
 - x. If *Accept* Message is not received, the check fails, and the test stops here.

- xi. If a *PS_RDY* Message is not received within *tPSTransition* min from the last bit of the EOP of *Accept* Message, the check fails, and the test stops here.
[TEST.PD.PS.SRC.2#9]
- d. The Tester performs the following checks:

Decrease the power drawn by the Tester to less than *pSnkStdby* 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.SRC.2#3]
 2. The Tester checks that the V_{BUS} transition meets *vSrcSlewPos* and *vSrcSlewNeg*. [TEST.PD.PS.SRC.2#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.2#5]
 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.SRC.2#6]
 5. The Tester checks that the UUT does not send *PS_RDY* before the V_{BUS} is within *vSrcNew* or *vPpsNew*. [TEST.PD.PS.SRC.2#7]

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* + 15ms for 10 times. Once the Tester has resent *PR_Swap* for 10 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_{BUS} 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_{BUS} and then it sends a *PS_RDY* Message, both within *tNewSRC* max of the last *GoodCRC* sent by the UUT.
10. 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]
11. The UUT is now a Sink, and the Tester is a Source.
 12. The Tester sends an *Accept* Message.
 - a. If a new PDO has been requested:
 - i. After a total of *tSrcTransition* min, check that power drawn by the UUT does not exceed *pSnkStdby* 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 power drawn by the UUT, measured in the 1ms before sending *PS_RDY*, does not exceed *pSnkStdby* max. [TEST.PD.PS.SRC.3#6]
 - iv. The Tester sends a *PS_RDY* Message to the UUT
 - b. If no new PDO requested,
 - i. After 315ms time period, (*tSrcTransition* max + *tSrcReady* max) check that the current drawn from V_{BUS} does not exceed the previously contracted current (100mA). [TEST.PD.PS.SRC.3#7]
 - ii. The Tester sends a *PS_RDY* Message to the UUT.
 13. 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
 14. The UUT is now a Sink, and the Tester is a Source.
 15. The Tester presents *SinkTxOk* if the test is in PD3 mode. The Tester waits 500ms to respond to messages from the UUT.
 16. 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 1st PDO of the *Sink_Capabilities* from the UUT
 17. The check fails if the UUT does not responds with a *Request* Message.
[TEST.PD.PS.SRC.3#9]
 18. The Tester sends an *Accept* Message.
 - a. If a new PDO has been Requested:

- i. After a total of *tSrcTransition* min, check that power drawn by the UUT does not exceed *pSnkStdby* max. [TEST.PD.PS.SRC.3#10]
- 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.

Check that the average power drawn by the UUT, measured in the 1ms before sending *PS_RDY*, does not exceed *pSnkStdby* max. [TEST.PD.PS.SRC.3#11]

- iii. The Tester sends a *PS_RDY* Message to the UUT
- b. If no new PDO Requested,
 - i. After 315ms time period, (*tSrcTransition* max + *tSrcReady* max) check that the current drawn from V_{BUS} does not exceed the contracted current. [TEST.PD.PS.SRC.3#12]
 - ii. The Tester sends a *PS_RDY* Message to the UUT.

19. 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]

20. The Tester sends a new *Source_Capabilities* Message with a single PDO

- a. 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

21. The Tester checks that it receives a *Request* Message [TEST.PD.PS.SRC.3#14]

22. The Tester sends an *Accept* Message.

- a. After a total of *tSrcTransition* min, check that power drawn by the UUT does not exceed *pSnkSusp* max. [TEST.PD.PS.SRC.3#17]
- b. The Tester a *PS_RDY* Message to the UUT.

23. 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]

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: N/A

Test Conditions:

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

Test Procedures:

1. 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.
3. The check fails if the UUT does not send *Sink_Capabilities* Message. [TEST.PD.PS.SNK.1#1]
4. The Tester sends a new *Source_Capabilities* Message with PDOs, mirroring the VIF definition of the UUT Sink requirements (for Rev 3 include any PPS PDOs)
 - a. The first PDO:
 - i. B31...30 (Fixed Supply) set to 00b
 - ii. B29 (Dual-Role Power) set to 0b
 - iii. B28 (USB Suspend Supported) set to 0b
 - iv. B26 (USB Communications Capable) set to 0b
 - v. B25 (Dual-Role Data) set to 0b
 - 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 1st PDO of the *Sink_Capabilities* from the UUT

- b. The remaining PDOs mirror the VIF definition of the UUT Sink requirements (for Rev 3 include any PPS PDOs) If the *Sink_Capabilities* consists of only one PDO, the Tester provides only 1 PDO in the *Source_Capabilities*.
 5. The check fails if the UUT does not responds with a *Request* Message. [TEST.PD.PS.SNK.1#2]
 6. The Tester sends an *Accept* Message.
 - a. If a new PDO has been requested:
 - i. After a total of *tSrcTransition* min, check that power drawn by the UUT does not exceed *pSnkStdbby* max. [TEST.PD.PS.SNK.1#3]
 - ii. Wait a nominal *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. As soon as the voltage on V_{BUS} has stopped changing, check that power drawn by the UUT does not exceed *pSnkStdbby* max. [TEST.PD.PS.SNK.1#4]
 - iv. The Tester sends a *PS_RDY* Message to the UUT
 - b. If no new PDO requested,
 - i. After 315ms time period, (*tSrcTransition* max + *tSrcReady* max) check that the current drawn from V_{BUS} does not exceed the previously contracted current (100mA). [TEST.PD.PS.SNK.1#5]
 - ii. The Tester sends a *PS_RDY* Message to the UUT.
7. The Tester performs the following check [TEST.PD.PS.SNK.1#6]
 - 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
8. The Tester sends another new *Source_Capabilities* Message with single PDO
 - a. The PDO:
 - i. B31...30 (Fixed Supply) set to 00b
 - ii. B29 (Dual-Role Power) set to 0b
 - iii. B28 (USB Suspend Supported) set to 0b
 - iv. B26 (USB Communications Capable) set to 0b
 - v. B25 (Dual-Role Data) set to 0b
 - 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
9. The check fails and the test ends here if the UUT does not respond with a *Request* Message [TEST.PD.PS.SNK.1#7]
10. The Tester sends an *Accept* Message.
 - a. If a new PDO has been requested:

- i. After a total of *tSrcTransition* min, check that power drawn by the UUT does not exceed *pSnkStdby* max. [TEST.PD.PS.SNK.1#8]
 - ii. Wait a nominal *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. As soon as the voltage on V_{BUS} has stopped changing, check that power drawn by the UUT does not exceed *pSnkStdby* max. [TEST.PD.PS.SNK.1#9]
 - iv. The Tester sends a *PS_RDY* Message to the UUT
 - b. If no new PDO requested,
 - i. After a total of *tSrcTransition* min, check that power drawn by the UUT does not exceed *pSnkSusp* max. [TEST.PD.PS.SNK.1#10]
 - ii. The Tester sends a *PS_RDY* Message to the UUT.
11. 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#11]

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
<i>Rev2Snk</i>	✓	
<i>Rev3ChkdSnk</i>	✓	

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* + 15ms for 10 times. Once the Tester has repeated for 10 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 12**), based on the Source PDOs specified in the VIF:
 - a. The Tester defines the following *Request* Message:
 - i. B27 (GiveBack Flag) = 0b
 - ii. B26 (Capability Mismatch) = 0b
 - iii. B25 (USB Communication Capable) = 0b
 - iv. B24 (No USB Suspend) = 1b
 - v. B19...10 (Operating Current / Power) = 100mA / 0.5W
 - vi. B9...0 (Maximum Operating Current / Power) = Maximum Current / Power in the Source PDO.
 - 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 10 times or until *Accpet* 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 a *PS_RDY* Message is not received within *tPSTransition* min from the last bit of the EOP of *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 10 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 a *PS_RDY* Message is not received within *tPSTransition* min from the last bit of the EOP of *Accept* Message, the check fails, and the test stops here. [TEST.PD.PS.SNK.2#12]
- v. The Tester responds to any *Get_Sink_Capabilities* 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 14).
- vi. If Source Capabilities are received with less than the Sink Capabilities (same as VIF Source Capabilities) and 2 seconds have not elapsed, then the Tester goes back and repeats the *Request* Message in step I with the Mismatch bit set.
- vii. 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.SNK.2#13]
- viii. When the required Source Capabilities have been received by the Tester, it sends *Request* Message defined above.
- ix. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of 10 times or until *Accept* is received.
- x. If *Accept* Message is not received, the check fails, and the test stops here.
- d. The Tester performs the following checks:

Decrease the power drawn by the Tester to less than *pSnkStdby* 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]
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 or *vPpsNew* min limit shall be (*vSrcNew* min) - $(0.75 \cdot I/X)$ or (*vPpsNew* min) - $(0.75 \cdot I/X)$, 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* + 15ms for 10 times. Once the Tester has repeated for 10 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_{BUS} 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_{BUS} .) The Tester draws the requested Operating Current / Power with a current transition slew rate of 100mA per μ s. The Tester monitors V_{BUS} voltage and the check fails if V_{BUS} voltage is not within the limits of PpsNew. [TEST.PD.PS.SNK.3#4]
 - b. The Tester defines *Request* Message as following:
 - i. B27 (GiveBack Flag) = 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. B19...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.

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 10 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 *tPSTransition* min from the last bit of the EOP of *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 10 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 a *PS_RDY* Message is not received within *tPSTransition* min from the last bit of the EOP of *Accept* Message, the check fails, and the test stops here.
[TEST.PD.PS.SNK.3#13]
 - v. The Tester responds to any *Get_Sink_Capabilities* 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 14).
 - vi. If Source Capabilities are received with less than the Sink Capabilities (same as VIF Source Capabilities) and 2 seconds have not elapsed, then the Tester goes back and repeats the *Request* Message in step I with the Mismatch bit set.
 - vii. 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.SNK.2#14]
 - viii. When the required Source Capabilities have been received by the Tester, it sends *Request* Message defined above.
 - ix. If *Wait* Message is received, the Tester re-sends *Request* Message every *tSinkRequest* up to a maximum of 10 times or until *Accept* is received.
 - x. If *Accept* Message is not received, the check fails, and the test stops here.
- 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 μ 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 μ 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, decrease the power drawn by the Tester to less than *pSnkStdby* 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 that V_{BUS} remains 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.3#6]
 2. The Tester checks that the V_{BUS} transition meets *vSrcSlewPos* and *vSrcSlewNeg*. [TEST.PD.PS.SNK.3#7]
 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* 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* before the V_{BUS} is within *vSrcNew* or *vPpsNew*. [TEST.PD.PS.SNK.3#10]
 6. After having sent *GoodCRC* in response to *PS_RDY*, the Tester increases its current draw to:
 - a. For Fixed, Battery and Variable 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* 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 (6 separate *Request* Messages for PPS):
 - 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 PDO, replace B19...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
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 performs below [TEST.PD.EPR.SRC3.1#2]
 - a) If the VIF field EPR_Supported_As_Src is Yes, The Tester checks that 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 UUT responds with *Not_Supported* Message and the test stops here.
4. If the VIF field *Captive_Cable* is No, Tester performs below checks
 - a. Tester checks that UUT does not send a *VCONN_Swap* message [TEST.PD.EPR.SRC3.1#3]
 - b. The check fails if the UUT does not send SOP' *Discover Identity* Command
[TEST.PD.EPR.SRC3.1#4]
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* (500ms), 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]
6. 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]
7. Tester checks that the UUT establishes EPR contract using common procedure
COMMON.PROC.PD3.2 [TEST.PD.EPR.SRC3.1#10]
8. 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.PD2.2, 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.SRC3.1#7]
 - c. If the UUT responds with *Wait* Message, the Tester resends *Request* Message every *tSinkRequest* for 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries, and the test stops here. [TEST.PD.EPR.SRC3.1#12]
 - d. The check fails if *PS_RDY* Message is not received within *tPSTransition* min(EPR mode) from the last bit of the EOP of *Accept* Message. [TEST.PD.EPR.SRC3.1#8]
9. 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
 10. 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]
 11. 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 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. The Tester checks that UUT sends a *VCONN_Swap* message [TEST.PD.EPR.SRC3.2#5], else the check fails, and test stops here

9. The Tester checks that UUT completes the Vconn swap sequence using COMMON.PROC.PD.6
10. 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)
11. The check fails if the UUT does not send a SOP' *Discover Identity* Command.
[TEST.PD.EPR.SRC3.2#6]
12. 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 (500ms), 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. 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]
14. Tester checks that the UUT establishes EPR contract using common procedure COMMON.PROC.PD3.2 [TEST.PD.EPR.SRC3.2#9]
15. 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* Message is not received. [TEST.PD.EPR.SRC3.2#10]
 - c. The check fails if *PS_RDY* Message is not received within tPSTransition min(EPR mode) from the last bit of the EOP of *Accept* Message. [TEST.PD.EPR.SRC3.2#11]
16. 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
17. 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]
18. 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 Source_Caps 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 EPR_Source_Caps 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 or 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 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. The Tester checks that UUT sends a *VCONN_Swap* message [TEST.PD.EPR.SRC3.4#5], else the check fails, and test stops here
9. If the UUT initiates the *VCONN_Swap* Message, Tester responds with *Reject* Message
10. 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*(500ms) (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		✓	✓

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.SRC3.6#2]
4. If a SOP' *Discover Identity* Command is received from the UUT, the Tester replies as a Passive Cable with
 - a. V_{BUS} Current Handling Capability set to 10b(5A),
 - b. Maximum V_{BUS} Voltage set to 11b(50V),
 - c. EPR Mode Capable field set to 0b (EPR Mode Capable)
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*(500ms) (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. Repeat this test by Tester emulating as a Passive Cable
 - a. V_{BUS} Current Handling Capability set to 10b(5A),
 - b. Maximum V_{BUS} Voltage set to 00b(20V),
 - c. EPR Mode Capable field set to 1b (EPR Mode Capable)
9. Repeat this test by Tester emulating as a Passive Cable
 - a. V_{BUS} Current Handling Capability set to 01b(3A),
 - b. Maximum V_{BUS} Voltage set to 11b(50V),
 - c. EPR Mode Capable field set to 1b (EPR Mode Capable)

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 just after 25us from the *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		✓	✓

- There are 2 possible bring-up procedures:
 - 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).
 - 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 an *EPR_Mode* (Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)
- 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]
- 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]
- 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)
- The check fails if the UUT does not send SOP' *Discover Identity* Command[TEST.PD.EPR.SRC3.8#4]
- 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* (500ms), 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_{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 check fails if the UUT does not send SOP' *Discover Identity* Command [TEST.PD.EPR.SRC3.9#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* (500ms), 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. 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_Cap* 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 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries, and the test stops here [TEST.PD.EPR.SRC3.9#14]
 - d. The check fails if *PS_RDY* Message is not received within *tPSTransition* min(EPR mode) from the last bit of the EOP of *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. Tester sends a *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_Cap* message.
 - b. The check fails if the VIF field *EPR_Supported_As_Src* is set to No and the UUT does not respond with a *Not_Supported* Message and the tests stops here.
4. 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 send 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_{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 check fails if the UUT does not send SOP' *Discover Identity* Command [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 (500ms), 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:

- There are 2 possible bring-up procedures:
 - 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 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 an *EPR_Mode* (Enter) Message with the Action field set to 0x01(Enter) and Data field set to 140 (140W)
- 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]
- 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]
- 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)
- The check fails if the UUT does not send SOP' *Discover Identity* Command [TEST.PD.EPR.SRC3.12#4]
- 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 (500ms), 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 send 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_{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 check fails if the UUT does not send SOP' *Discover Identity* Command [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* (500ms), 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]

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:

- 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.PS.EPR.SRC3.1#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.PS.EPR.SRC3.1#2]
- The Tester repeats the following steps for each of the EPR_PDOs only (in the *EPR_Source_Capabilities* Message), starting with the 8th PDO:
 - The Tester sends an *EPR_Request* Message.
 - B27 (GiveBack Flag) = 0b
 - B26 (Capability Mismatch) = 0b
 - B25 (USB Communication Capable) = 0b
 - 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 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 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries, and the test stops here. [TEST.PD.PS.EPR.SRC3.1#13]
- d. The check fails if *PS_RDY* Message is not received within *tpSTransition* min from the last bit of the EOP of *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 μ 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 μ 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, decrease the power drawn by the Tester to less than $p_{SnkStdby}$ within $t_{SnkStdby}$ of the last bit 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 v_{SrcNew} or v_{AvsNew} of the starting voltage within $t_{SrcTransition}$ 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 $v_{SrcSlewPos}$ and $v_{SrcSlewNeg}$. [TEST.PD.PS.EPR.SRC3.1#6]
 3. During the voltage transition, the Tester checks that the V_{BUS} voltage remains within $v_{SrcValid}$ limits from the time of crossing into the $v_{SrcValid}$ limits until $t_{SrcSettle}$ max. [TEST.PD.PS.EPR.SRC3.1#7]
 4. The Tester checks that the V_{BUS} voltage remains within v_{SrcNew} or v_{AvsNew} limits between $t_{SrcSettle}$ max and 100ms after $t_{SrcSettle}$ 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 v_{SrcNew} or v_{AvsNew} . [TEST.PD.PS.EPR.SRC3.1#9]
 6. After having sent *GoodCRC* in response to *PS_RDY*, the Tester increases its current draw to the current requested and The Tester checks that the V_{BUS} voltage remains within v_{SrcNew} or v_{AvsNew} 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
Rev3ChkdSrc		✓

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.PS.EPR.SRC3.2#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.PS.EPR.SRC3.2#2]
- NOTE*
The Tester repeats the following steps using the PDO as provided in the EPR_PDO Sequence Table (see [Table 13](#)). : 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.
 - The Tester sends an *EPR_Request* Message:
 - B27 (GiveBack Flag) = 0b
 - B26 (Capability Mismatch) = 0b
 - B25 (USB Communication Capable) = 0b
 - B24 (No USB Suspend) = 1b
 - B22 (EPR Mode Capable) = 1b
 - 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 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries. [TEST.PD.PS.EPR.SRC3.2#10]
- d. The check fails if *PS_RDY* Message is not received within *tPSTransition* min from the last bit of the EOP of *Accept* Message. [TEST.PD.PS.EPR.SRC3.2#4]
- e. The Tester performs the following checks:

Decrease the power drawn by the Tester to less than *pSnkStdby* 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]
- 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_Caps 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. Tester checks that the “EPR mode capable” field in the *Request* Message sent by UUT matches with VIF parameter EPR_Supported_As_Snk during the bring up procedure
[TEST.PD.EPR.SNK3.1#1]
3. Tester waits for 5 second 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_{BUS} Current Handling Capability set to 10b(5A)
 - f. Maximum V_{BUS} Voltage set to 11b(50V)
 - g. EPR Mode Capable field set to 1b (EPR Mode Capable)
5. Tester sends `EPR_Mode` Message with Action field set to 0x03(Enter_Succeeded) and Data field set to 0x0 within `tEnterEPR` from the `EPR_Mode` (Enter) Message.
6. Tester sends `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. Tester waits for 10 seconds and if there are no messages initiated by the UUT for more than `tSinkEPRKeepAlive` max (500ms), then the check fails [TEST.PD.EPR.SNK3.1#4]
8. If the UUT does not send even single `EPR_KeepAlive` message in above step, make this check as warning [TEST.PD.EPR.SNK3.1#5]
9. Tester waits for 1 second and if the UUT does not send `EPR_Request` Message with SPR RDO, the Tester sends `EPR_SourceCaps` message with no EPR PDO's and checks that the UUT responds with `EPR_Request` Message [TEST.PD.EPR.SNK3.1#6]
10. Tester checks that the UUT establishes the EPR contract negotiation using COMMON.PROC.PD3.4 [TEST.PD.EPR.SNK3.1#8]
11. 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
12. Tester sends `SourceCaps` message and checks that the UUT responds with `Request` Message [TEST.PD.EPR.SNK3.1#7]
13. 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 *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 *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_{BUS} Current Handling Capability set to 10b(5A)
 - b. Maximum V_{BUS} Voltage set to 11b(50V)
 - c. EPR Mode Capable field set to 1b (EPR Mode Capable)
6. Tester doesn't send *EPR_Mode* (Enter_Succeeded) Message.
7. Tester checks that UUT sends *Soft Reset* within the *tEnterEPR*(500ms) from the EOP of *GoodCRC* Message of the *EPR_Mode* (Enter) Message.[TEST.PD.EPR.SNK3.2#4]

TEST.PD.EPR.SNK3.3 EPR Entry Failed by *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 *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* of *EPR_Mode* (Enter Failed) Message to the last bit of the *Soft Reset* Message). [TEST.PD.EPR.SNK3.3#3]
6. If UUT initiates *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 *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 *EPR_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.
5. Tester sends *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 sends a *Hard Reset* Signaling within *tTypeCSinkWaitCap* (this timing is measured from the last bit of the *GoodCRC* 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 *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 *EPR_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.
5. Tester sends *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, PDO2- 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* of 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_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 *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 *EPR_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.
5. Tester sends *EPR_Mode* Message with Action field set to 0x03(Enter_Succeeded) after 100ms from the *EPR_Mode* (Enter) Message.
6. Tester sends *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 EPR_KeepAlive message within *tSinkEPRKeepAlive* max(500ms) 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 filed 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* of *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

Description: The Tester as a source verifies the UUT responds with *Hard Reset* for the *Wait* 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.SNK3.7#1]
3. Tester waits for 5 seconds for the UUT to initiate *EPR_Mode* (Enter) Message with the Action field set to 0x01(Enter) , otherwise the test passes and stops here [TEST.PD.EPR.SNK3.7#2]
4. Tester responds with *EPR_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.
5. Tester sends *EPR_Mode* Message with Action field set to 0x03(Enter_Succeeded) after 100ms from the *EPR_Mode* (Enter) Message.
6. Tester initiates the EPR_SourceCaps message after 50ms from the *EPR_Mode* (Enter_Succeeded) Message
7. Tester checks that the UUT sends *EPR_Request* Message for which the Tester responds with *Wait* Message [TEST.PD.EPR.SNK3.7#3]
8. Tester checks that the UUT sends a *Hard Reset* Signaling within *tProtErrHardReset*(this timing is measured from the last bit of the *GoodCRC* of *Wait* Message to the last bit of the *Hard Reset* Signal) [TEST.PD.EPR.SNK3.7#4]

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 *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 *EPR_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.
5. Tester sends *EPR_Mode* Message with Action field set to 0x03(Enter_Succeeded) after 100ms from the *EPR_Mode* (Enter) Message.
6. Tester sends *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(500ms) 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* of Source Cap 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:

- 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 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]
- Tester waits for 5 second for the UUT to initiate *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]
- Tester responds with *EPR_Mode* Message with the Action field set to 0x02(Enter Acknowledged) and Data field set to 0x0.
- Tester sends *EPR_Mode* Message with Action field set to 0x03(Enter_Succeeded) after 100ms from the *EPR_Mode* (Enter) Message.
- Tester initiates the SourceCaps message after 50ms from the *EPR_Mode* (Enter_Succeeded) Message
- Tester checks that the UUT sends a *Hard Reset* Signaling within *tProtErrHardReset*(this timing is measured from the last bit of the *GoodCRC* of *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
Rev3ChkdSnk	✓		✓
Rev3UnchkdSnk	✓		✓

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 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]
- Tester waits for 5 seconds for the UUT to initiate *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]
- 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 *EPR_Request* Message, else go directly to step 6. [TEST.PD.EPR.SNK3.10#3]
- Tester checks that the UUT establishes the EPR contract negotiation using COMMON.PROC.PD3.4 [TEST.PD.EPR.SNK3.10#4]
- 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 *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* of *EPR_Mode* Exit message to the last bit of the *Hard Reset* Signal. [TEST.PD.EPR.SNK3.10#6]

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 7-1.

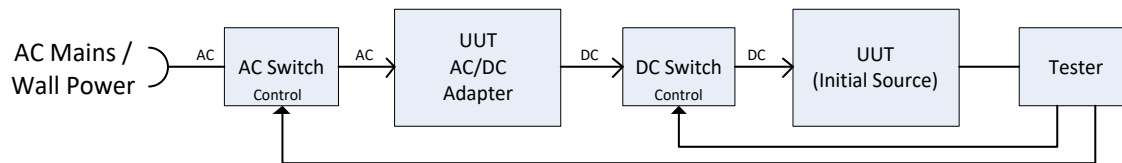


Figure 7 FRS Testing Concept

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 4](#)).

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*.

	Consumer Only, C/P, P/C, DRP Provider Only,	
<code>Rev3ChkdSrc5VAC</code>		✓
<code>Rev3ChkdSrc5VDC</code>		✓
<code>Rev3ChkdSrcHiVAC</code>		✓
<code>Rev3ChkdSrcHiVDC</code>		✓

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. The Tester operates the appropriate power disconnection switch (see Appendix B) depending on the Test Condition.
6. 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.
7. 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 *pSnkStdby*, 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 *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].

8. 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 v_{Safe5V} max
 - iii. The Tester applies v_{Safe5V} on V_{BUS} at $t_{SrcFRSwap}$ max after receiving the Fast Role Swap signal and V_{BUS} has dropped below v_{Safe5V} 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 $t_{Receive}$ max + $t_{ProtErrHardReset}$ 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_{BUS} electrical and PD messaging test sub-steps are performed separately.
 - b. V_{BUS} electrical sub-steps:
 - iv. The Tester immediately stops sinking current through V_{BUS} after detecting Fast Role Swap signal
 - v. The Tester waits for V_{BUS} to be below $vSafe5V$ max
 - vi. 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.
 - 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 PS_RDY Message is received. [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 15ms 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,	C/P, P/C, DRP
----------------	---------------

	Provider Only,	
Rev3ChkdSrc5VAC		✓
Rev3ChkdSrc5VDC		✓

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 a *PS_RDY* at the deadline limit of *tPSSourceOn* min after the time the last bit of the EOP of the *GoodCRC* Message corresponding to the received *PS_RDY* Message was sent.
 - 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
Rev3ChkdSrc5VAC		✓
Rev3ChkdSrc5VDC		✓

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 Provider/Consumer, or DRP, the Tester runs bring-up procedure with the UUT as a Source COMMON.PROC.BU.1.
- 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.
- After receiving the Fast Role Swap signal, the V_{BUS} electrical and PD messaging test sub-steps are performed separately.
 - V_{BUS} electrical sub-steps:
 - The Tester immediately stops sinking current through V_{BUS} after detecting Fast Role Swap signal
 - The Tester waits for V_{BUS} to be below *vSafe5V* max
 - 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.
 - PD messaging sub-steps:
 - 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]
 - Upon receipt of the *Accept* Message, the Tester replies with a *GoodCRC* Message.
 - Upon receipt of the *PS_RDY* Message, the Tester replies with a *GoodCRC* Message.
 - 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

	P/C, C/P, DRP	Consumer Only, Provider Only
<i>Rev3ChkdSnk5V</i>	✓	
<i>Rev3ChkdSnkHiV</i>	✓	

Test Procedures:

1. For Test Condition *Rev3ChkdSnkHiV*, during step 2 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. 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. Tester waits 500ms 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]

4. The Tester sends a Fast Role Swap signal to the UUT, with a duration of $t_{FRSwapTx}$ min. The signal is applied via a resistance of $r_{FRSwapTx}$ (5 Ω) to ground.
5. 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 10 μ F 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 $FR_Swap_Type_C_Current_Capability_As_Initial_Sink$. After the Tester has turned off V_{BUS} , the operating point when V_{BUS} is below v_{Safe5V} max is referred to as *Stage 1*.
 - iii. The Tester makes the following check on V_{BUS} :
 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_{BUS} 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_{BUS} and it reaches v_{Safe5V} min within $t_{SrcFRSwap}$ of both the Fast Role Swap signal having had time to be detected, and V_{BUS} being below v_{Safe5V} min (due to Tester turning off V_{BUS}). [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 $t_{FRSwapRx}$ max plus $t_{FRSwapInit}$ 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 $t_{ReceiverResponse}$ 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 $t_{FRSwapComplete}$ 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 $t_{FRSwapComplete}$ 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
Rev3ChkdSnk	✓	

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 500ms 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_{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 attempts to draw 100mA. (This is necessary in order 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 *vSafe5V* 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
Rev3ChkdSnk	✓	

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`.
 - 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.SNK3.3#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 15ms 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
Rev3ChkdSnk	✓	

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.
 - 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 max (see Table 16). 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
Rev3ChkdSnk	✓	

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.SNK3.5#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 R_p to R_d as soon as the operating point has reached Stage 1.
 - v. The Tester sends *PS_RDY* at the deadline limit of *tPSSourceOff* min. The delay is measured from the time the last bit of the EOP of the *GoodCRC* Message corresponding to the *Accept* Message was sent to the time the last bit of the EOP of the *PS_RDY* was sent.
 - 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
<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_{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 changes from presenting Rp to Rd as soon as the operating point has reached Stage 1.
- 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 *PS_RDY* Not Sent

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_{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 v_{Safe5V} 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 $t_{FRSwapRx}$ max plus $t_{FRSwapInit}$ 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 $t_{PSSourceOn}$ 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 *PS_RDY* is received. [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 15ms 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 command 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 “Data_Reset” message to UUT
3. The Tester checks for response.
 - a. If the tester receives message ‘*Not_Supported*’ and the VIF field ‘Data_Reset_Supported=No’, the check passes and test ends here. [TEST.PD.USB4.DRST.1#1]
 - b. If the tester doesn’t receive “*Not_Supported*” and the VIF field ‘Data_Reset_Supported=No’, the check fails and test ends here. [TEST.PD.USB4.DRST.1#2]
 - c. If the tester doesn’t receive *Accept* and the VIF Field Data_Reset_Supported= Yes. The check fails and test end here [TEST.PD.USB4.DRST.1#3]
 - d. If the tester receives *Accept* tester perform below step
 - a. Tester Power Cycle Vconn in accordance with requirements
4. The Tester sends the message ‘Data_Reset_Complete’ and verify the below checks upon the *GoodCRC* from UUT [TEST.PD.USB4.DRST.1#4]
 - a. No disconnect seen in CC termination.
 - b. UFP is not presenting Vconn
5. The Tester sends the *Get_Sink_Cap* Message and verify the below checks upon the *GoodCRC* from UUT [TEST.PD.USB4.DRST.1#5]
 - a. Data_role is same as before data reset (Tester is DFP, UUT is UFP)

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.

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 “Data_reset” message to UUT
3. The Tester checks for response.
 - a. If the tester doesn’t receive *Accept*, the check fails. [TEST.PD.USB4.DRST.2#1]
 - b. If the tester receives *Accept*, tester perform below steps [TEST.PD.USB4.DRST.2#2]
 - a. Power Cycle Vconn in accordance with requirements
4. The Tester skips sending “DataResetComplete” and sends the message “Get Sink cap” after *tdataresetfail* (300mS). Tester verifies that the UUT does error recovery in 315mS. [TEST.PD.USB4.DRST.2#3]

TEST.PD.USB4.DRST.3 –Data_Reset command response of UFP UUT Sourcing Vconn

Description:

Verify the Vconn power cycling of UFP upon accepting data_reset command 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 *tVCONNSwapWait* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *VCONN_Swap* for 10 times.
 - b. The check fails and test ends here if the UUT does not send an *Accept* Message.
3. Tester send message "Data_Reset" message to UUT
4. The Tester checks for response.
 - a. If the tester doesn't receive *Accept*, the check fails. [TEST.PD.USB4.DRST.3#2]
 - b. If the tester receives *Accept*, tester perform below steps
 - I. Check fails if the Vconn voltage is not below *vRaReconnect* (800mV) in *tVconnzero* (125mS) [TEST.PD.USB4.DRST.3#3]
 - II. Check fails if the *PS_RDY* Message is not received within minimum of *tVCONNSourceDischarge* [160mS]. [TEST.PD.USB4.DRST.3#4]
 - III. Check fails if the *PS_RDY* Message is received before VCONN voltage is below *vRaReconnect* (800mV) [TEST.PD.USB4.DRST.3#7]
5. The Tester sends the message Data_Reset_Complete and verify the below checks upon the *GOODCRC* from UUT
 - a. No disconnect seen in CC termination. [TEST.PD.USB4.DRST.3#5]
 - b. UFP not presenting Vconn. [TEST.PD.USB4.DRST.3#6]

TEST.PD.USB4.DRST.4 –DataReset 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. Tester sends *VCONN_Swap* message to bring UFP UUT as Vconn provider. Tester check 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* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *VCONN_Swap* for 10 times.
 - b. The check fails and test ends here if the UUT does not send an *Accept* Message.
3. Tester send message “Data_Reset” message to UUT
4. The Tester checks for response. [TEST.PD.USB4.DRST.4#2]
5. The Tester sends message *VCONN_Swap* without waiting (in less than 160mS after *Accept*) for *PS_RDY* from UUT and verify that the UUT does the error recovery in 15mS. The check fails if the error recovery is not seen from UUT. [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 data_reset command and provides the proper response.

Verify the Vconn power cycle sequence and DFP becomes the Vconn 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 message “Data_Reset” message to UUT. The Tester checks for response.
 - a. If the tester doesn’t receive *Accept* the check fail. Test end here. [TEST.PD. USB4.DRST.5#1]
 - b. If the tester receives *Accept*, the tester performs below steps
 - I. Tester verifies the UUT Power Cycle the Vconn properly. The UUT should stop sourcing Vconn and the VCONN should be below *vRaReconnect* (800mV) within *tVconnzero*(125mS). [TEST.PD. USB4.DRST.5#2]. The *tVconnzero* is measured from last bit of *GoodCRC* acknowledging *Accept*.
 - II. Tester verify the UUT keeps the Vconn below *vRaReconnect* (800mV) for *tVconnReapplied* (minimum of 10ms and max of 20mS). [TEST.PD. USB4.DRST.5#3]
 - III. Tester verifies that the UUT starts sourcing the Vconn (Vconn voltage rising above *vRaReconnect* by end of *tVconnReapplied*. [TEST.PD. USB4.DRST.5#4]
 - IV. The tester verifies the Vconn voltage is reaching in valid range in Max of *tVconnvalid* [5mS].[TEST.PD. USB4.DRST.5#5]. The time is measured between Vconn voltage crossing *vRaReconnect*(800mV) to min of *vVconnValid*(3V).

3. The Tester expects the Data_Reset_Complete message from UUT. If the Data_Reset_Complete message is received, and the tester considers the data_reset process is completed. Tester verifies the below checks

- a. UUT not initiated any other message after accepting data_reset. [TEST.PD. USB4.DRST.5#6]
- b. V_{BUS} is not changed, and it is in valid range throughout the data_reset process. [TEST.PD. USB4.DRST.5#7]

4. If the tester doesn't receive the Data_Reset_Complete in *tDataResetFail*(300mS) time, the check fails. [TEST.PD. USB4.DRST.5#8].

5. If the tester doesn't receive the Data_Reset_Complete in *tDataResetFail*(300mS) time, tester verifies that the UUT enters error recovery. Check fails if UUT not entered error recovery [TEST.PD. USB4.DRST.5#9]

6. If the VIF "USB4_Supported= Yes", Tester expects the UUT to initiate the device discovery process from the UUT. Tester expects DiscoverID REQ from UUT on SOP [TEST.PD. USB4.DRST.5#10]

7. If the VIF "USB4_Supported= Yes", Tester expects the UUT to initiate the cable discovery process from the UUT. Tester expects DiscoverID REQ from UUT on SOP' [TEST.PD. USB4.DRST.5#11]

TEST.PD. USB4.DRST.6 –Data_Reset command response of DFP UUT, UFP Sourcing Vconn

Description:

The Tester verifies DFP UUT accepts data_reset command and provides the proper response when UFP sourcing Vconn.

Verify the Vconn power cycle sequence and DFP becomes the Vconn 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* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *VCONN_Swap* for 10 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
4. The Tester send message “Data_reset” message to UUT. The Tester checks for response.
 - a. If the tester doesn’t receive *Accept*, the check fails. [TEST.PD. USB4.DRST.6#2]
 - b. If the tester receives *Accept* Message, the tester performs below steps
 - I. Tester sends *PS_RDY* after *tVconnZero*. The *tVconnzero* is measured from last bit of *GoodCRC* acknowledging *Accept*.
 - II. Tester verifies that the Vconn is below *vRaReconnect* for minimum of *tVconnReapplied*(10ms). The time is measured from the last bit of *GoodCRC* acknowledging *PS_RDY* to Vconn voltage rising above *vRaReconnect*. [TEST.PD. USB4.DRST.6#3]
 - III. Tester verifies that the UUT starts sourcing the Vconn (Vconn voltage rising above *vRaReconnect* by maximum of *tVconnReapplied*. [TEST.PD. USB4.DRST.6#4]
 - IV. The tester verifies the Vconn voltage is reaching in valid range in Max of *tVconnvalid* [5mS]. [TEST.PD. USB4.DRST.6#5]. The time is measured between Vconn voltage crossing *vRaReconnect*(800mV) to min of *vVconnValid*(3V).
3. The Tester expects the Data_Reset_Complete message from UUT. If the Data_Reset_Complete message is received, and the tester considers the data_reset process is completed. Tester verifies the below checks
 - a. UUT not initiated any other message after accepting data_reset. [TEST.PD. USB4.DRST.6#6]
 - b. *V_{BUS}* is not changed, and it is in valid range throughout the data_reset process. [TEST.PD. USB4.DRST.6#7]

4. If the tester doesn't receive the Data_Reset_Complete in *tDataResetFail*(300mS) time, the check fails. [TEST.PD. USB4.DRST.6#8].

6. If the VIF "USB4_Supported= Yes", Tester expects the UUT to initiate the device discovery process from the UUT. Tester expects DiscoverID REQ from UUT on SOP [TEST.PD. USB4.DRST.6#9]

7. If the VIF "USB4_Supported= Yes", Tester expects the UUT to initiate the cable discovery process from the UUT. Tester expects DiscoverID REQ from UUT on SOP' [TEST.PD. USB4.DRST.6#10]

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 data_reset command 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* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *VCONN_Swap* for 10 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 send message “Data_reset” message to UUT. The Tester checks for response.

- a. If the tester doesn’t receive *Accept*, the check fails. [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 the Vconn.
 - II. The Tester checks for UUT going error recovery upon expiry of Vconndisargetimer. The check fails if the error recovery is not seen in *tVCONNSourceDischarge* max (240mS).
[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
 - b. If the UUT responds with a *Wait* Message and the Tester resends *DR_Swap* every *tDRSwapWait* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *DR_Swap* for 10 times. [TEST.PD.USB4.EUSB.1#1]
 - c. If the UUT not responded with *Accept*. The check fails and test end here. [TEST.PD.USB4.EUSB.1#2]
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 10 times. The check fails if the Tester has resent *Discover Identity* Command for 10 times.
 - d. The check fails if the UUT responds with a “Responder BUSY” message and the Tester has resent *Discover Identity* Command for 10 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: USB3.2 Gen2, 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 *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 *Wait* Message, the Tester resends Request Message every tSinkRequest for 10 times or until *Accept* Message is received. The check fails if the UUT responds with Wait Message after 10 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(500mS). For Consumer Only/C/P and the DRP the tEnter USB 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: USB3.2 Gen2, 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
 - II. If the UUT responds with a *Wait* Message and the Tester resends *DR_Swap* every *tDRSwapWait* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *DR_Swap* for 10 times. [TEST.PD.USB4.EUSB.2#1]
 - III. If the UUT not responded with *Accept*. The check fails and test end here. [TEST.PD.USB4.EUSB.2#2]

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 10 times. The check fails if the Tester has resent *Discover Identity* Command for 10 times.
- i. The check fails if the UUT responds with a “Responder BUSY” message and the Tester has resent *Discover Identity* Command for 10 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 information received from *Discover Identity* ACK response from UUT

USB Mode: USB4, 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*(500mS). For Consumer Only/C/P and the DRP the tEnter USB 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 Gen3 Active cable. Highest Speed USB4Gen3, Current – 3A, V_{BUS} 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= Tester Vendor ID. All other values are 0
 - Active Cable VDO 1-> Connector: USB TypeC, Max Current 3A, Max Voltage=20V, Max Speed= USB4 Gen3. Cable termination = 11b. Cable latency=0001b, SOP” Controller Present=1, V_{BUS} 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 Gen3
- 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 *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* + 15ms for 10 times. The check fails and the Test ends here if the Tester has resent *DR_Swap* for 10 times.
 - b. 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.

- 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*(500mS) [TEST.PD. USB4.EUSB.3#6] For Provider, DRP, P/C UUTs, the tEnter USB is measured from last bit of *GoodCRC* acknowledging source cap from the UUT. For C, C/P UUTs, the tEnter USB 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,
 - 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 USB 3 Gen 1 or above,
 - iii. PCIe Support field setting matches to the VIF field
USB4_PCl_e_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 (1000mS). 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 perform 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 Gen3, 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 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

- Active Cable VDO 1-> Connector: USB TypeC, Max Current 3A, Max Voltage=20V, Max Speed= USB4 Gen3, Cable latency=0001b. Remaining all field 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 Gen3
- 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
 - b. If the UUT responds with a *Wait* Message and the Tester resends *DR_Swap* every *tDRSwapWait* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *DR_Swap* for 10 times. [TEST.PD. USB4.EUSB.4#1]
 - c. If the UUT not responded with *Accept*. The check fails and test end here. [TEST.PD. USB4.EUSB.4#2]
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 10 times or until *Accept* Message is received. The check fails if the UUT responds with *Wait* Message after 10 retries [TEST.PD.USB4.EUSB.4#11].

4. Tester send *DR_Swap* Message to bring the UUT as DFP. Tester checks the response.
[TEST.PD.USB4.EUSB.4#4]

- a. If the UUT responds with a *Wait* Message and the Tester resends *DR_Swap* every *tDRSwapWait* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *DR_Swap* for 10 times.
- b. The check fails if the UUT not accepted the DR_SWAP, the test ends here.

5. If the tester receives the *Enter_USB* Message on SOP', the tester performs the below checks.
[TEST.PD.USB4.EUSB.4#5]

- a. Verify the USB mode on EUDO is set to USB4
- b. Tester send *Accept* response on SOP'

6. If the tester receives the *Enter_USB* Message on SOP'', the tester performs the below checks.
[TEST.PD.USB4.EUSB.4#6]

- a. Verify the USB mode on EUDO is set to USB4
- b. Tester sends *Accept* Message on SOP''

7. If the tester receives the *Enter_USB* Message on SOP, the Tester performs the below checks and sends *Accept* response if checks are true. Otherwise send *Reject* response. [TEST.PD.USB4.EUSB.4#7]

- a. The *Enter_USB* Message is received with in *tEnterUSB*(500mS). The *tEnter USB* is measured from last bit of *GoodCRC* acknowledging *Accept* Message in response for DR_SWAP
- b. Verify the USB mode on EUDO is set to USB4
- c. USB4 DRD on EUDO set to 1

8. Tester waits for *tUSB4Timeout* (1000mS). The check fails if the tester doesn't receive the *Enter_USB* on SOP. The *tUSB4Timeout* is measured *GoodCRC* of *Accept* for *DR_Swap*. [TEST.PD.USB4.EUSB.4#8]

9. Tester send *DR_Swap* Message to bring the UUT as UFP. Tester check the response.
[TEST.PD.USB4.EUSB.4#9]

- a. If the UUT responds with a *Wait* Message and the Tester resends *DR_Swap* every *tDRSwapWait* + 15ms for 10 times. The check fails and the test ends here if the Tester has resent *DR_Swap* for 10 times.
- b. The check fails if the UUT not accepted the DR_SWAP, the test ends here.

10. 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*. [TEST.PD.USB4.EUSB.4#10].

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:

- There are 3 possible bring-up procedures
 - 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
 - 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.
 - 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.
- 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 10 times every *tVDMBusy* min.
- 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.
 - If the VIF Field 'Enter_USB_Supported = No', the check fails if the tester receives any response message. [TEST.PD. USB4.CBL.1#1]
 - 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 DiscoverID 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 10 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 DiscoverID 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>1.3</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>
      </conditions>
    </test>
  </testRun>
</testRuns>

```

```

        </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.

The Tester Vendor shall provide Switch Boxes to accomplish these switching requirements.

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

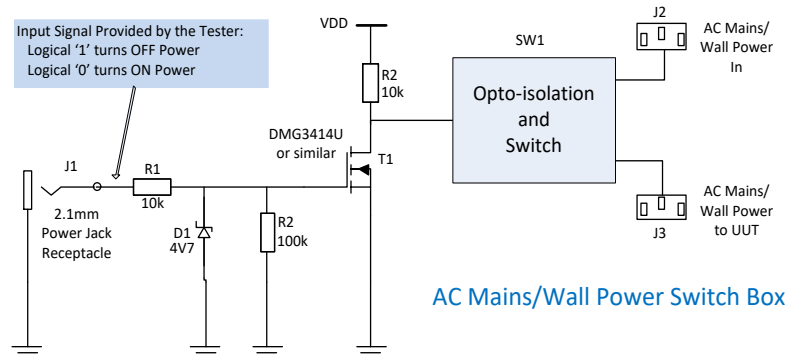


Figure 8 Schematic of the Switch Box for disconnecting AC Mains / Wall Power

Figure C-1 shows 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 components in the schematic as shown in Figure B-1:

- 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

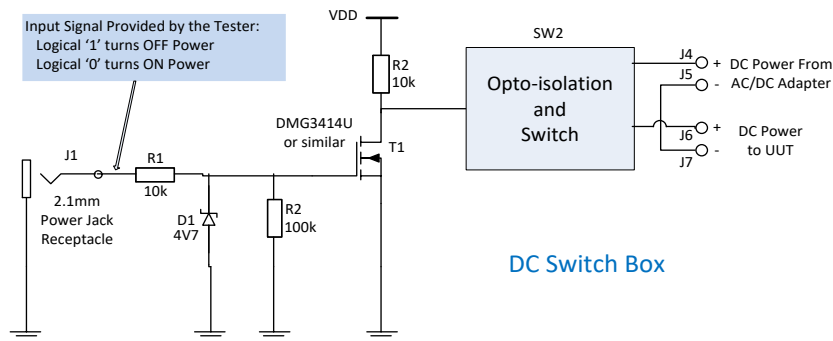


Figure 9 schematic of the standard switch box for disconnecting DC power

Most of the operation of the DC Switch Box (Figure C-2) 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 C-3: 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

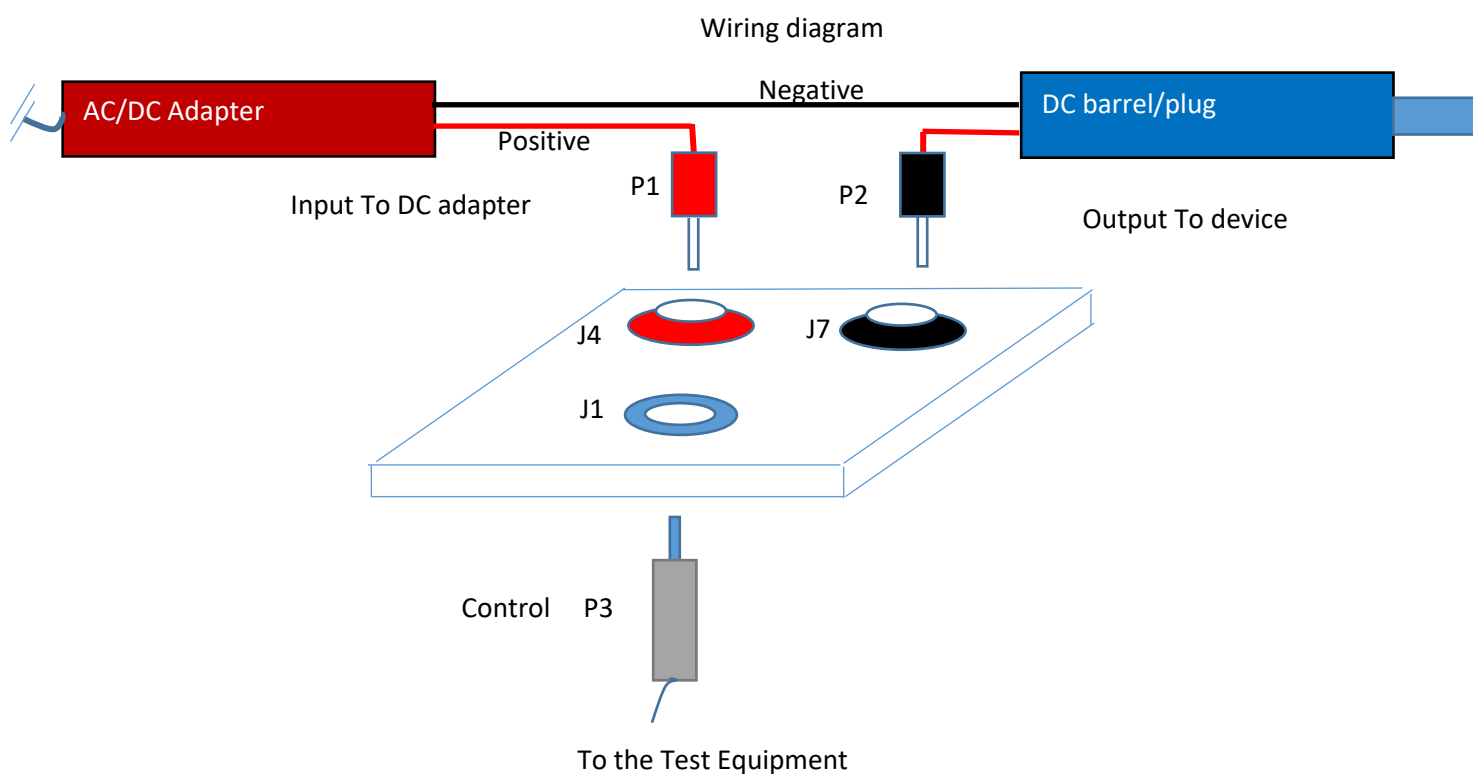


Figure 10 Wiring diagram

Table 15 Components

Component	Manufact.	Part Number	Newark #	Description
P1	Pomona	1325-2	30F1407	Conn banana Plug 15A, 60V Red
P2	Pomona	1325-0	30F1430	Conn banana Plug, 15A, 60V Black
P3	Switchcraft conxall	762Z	80AH1412	DC Power Connector
J1	Switchcraft Conxall	721A	49AC1396	DC Power Connector, 2.1mm Jack
J4, J7	2269-0	2269-0	30F1419	Banana Test Connector, 15A 25kV

Appendix D: V_{BUS} measurements

In order 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 20 μ s 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 15 μ s 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 10 μ s 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 2 μ s 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

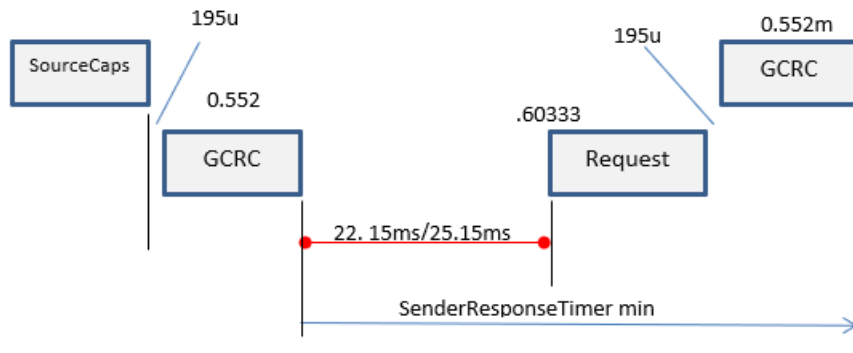
Table 16 Timing Table & Calculations

Timer	Spec Rev.	min	max	Reference
<i>tSenderResponse</i>	PD2	24 mS	30 mS	[PD2 Spec]
	PD3	27 mS	33 mS	[PD3 Spec]
<i>tCtsSnkRequest</i>	PD2	-	22.26 mS	Figure 11 tCtsSnkRequest Calculations
	PD3	-	25.26 mS	Figure 11 tCtsSnkRequest Calculations
<i>tCtsSrcAccept</i>	PD2	-	22.76 mS	Figure 12 tCtsSrcAccept Calculation
	PD3	-	25.76 mS	Figure 12 tCtsSrcAccept Calculation
<i>tCtsSenderResponse</i>		-	50mS	

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 11 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

$$24 - (0.49662 + 0.195 + 0.552) =$$

22.76ms

Latest valid time to start sending Request PD3

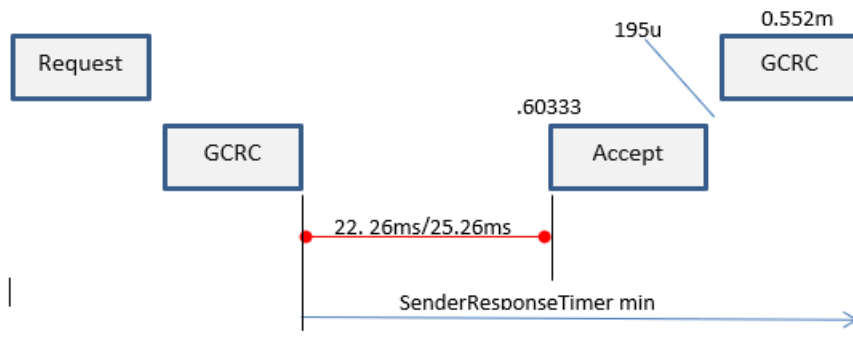
$$27 - (0.60333 + 0.195 + 0.552) =$$

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 12 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

$$24 - (0.60333 + 0.195 + 0.552) =$$

22.65ms

Latest valid time to start sending Request PD3

$$27 - (0.60333 + 0.195 + 0.552) =$$

25.65ms