# Universal Serial Bus Type-C™ Port Controller Interface Specification

**Revision 1.~~0~~ 0, Version 1.1**
**~~October~~ July ~~20,~~ 201~~5~~6**

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

**Copyright © 2015, USB 3.0 Promoter Group:**
**Hewlett-Packard Company, Intel Corporation, Microsoft Corporation,**
**Renesas, STMicroelectronics, and Texas Instruments**
**All rights reserved.**

LIMITED COPYRIGHT LICENSE: The USB 3.0 Promoters grant a conditional copyright license under the copyrights embodied in the USB Type-C Cable and Connector 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 of the 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 additional 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.

All implementation examples and reference designs contained within this Specification are included as part of the limited patent license for those companies that execute the USB 3.0 Adopters Agreement.

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

CONTENTS

Specification Work Group Chairs / Specification Editors ..... 10
Specification Work Group Contributors ..... 10
Revision, Version History ..... 12
1 Introduction ..... 13
1.1 Purpose ..... 13
1.2 Scope ..... 13
1.3 Related Documents ..... 14
1.4 Conventions ..... 14
1.4.1 Precedence ..... 14
1.4.2 Keywords ..... 14
1.4.3 Numbering ..... 15
1.5 Terms and Abbreviations ..... 15
2 Overview ..... 16
2.1 Introduction ..... 16
2.2 USB Type-C Port Controller (TCPC) Interface ..... 16
3 Type-C Port Controller Requirements ..... 17
3.1 Port Power Control for VBUS and VCONN ..... 17
3.2 USB CC Logic ..... 17
3.3 USB-PD Message Delivery ..... 17
3.4 Debug Accessory Detection ..... 18
3.5 State Diagram Introduction ..... 19
3.6 TCPC Protocol Layer State Operation ..... 19
3.7 USB Type-C Port Controller Requirements for Source, Sink, and DRP ..... 25
3.7.1 Source Requirements ..... 29
3.7.2 Sink Requirements: ..... 30
3.7.3 Sink with Accessory Support ..... 31
3.7.4 DRP Requirements ..... 32
3.8 Watchdog Timer Requirements (Optional Normative) ..... 33
3.8.1 Watchdog Timer Function ..... 33
4 USB Type-C Port Controller Interface ..... 34
4.1 Register Map ..... 35
4.2 TCPC SMBus Optional Normative Requirements ..... 38
4.3 Writing and Reading TCPC Registers ..... 38
4.3.1 Writing Single Byte Registers ..... 38
4.3.2 Reading Single Byte Registers ..... 39
4.3.3 Writing Multiple-Byte Registers ..... 39
4.3.4 Reading Multiple-Byte Registers ..... 40
4.3.5 Writing the TRANSMIT_BUFFER ..... 40
4.3.6 Reading the RECEIVE_BUFFER ..... 41
4.4 Register Definition ..... 42
4.4.1 Identification Registers ..... 42

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

4.4.2 ALERT Register (Required) ..... 44
4.4.3 Mask Registers ..... 46
4.4.4 CONFIGURE STANDARD OUTPUT (Optional Normative) ..... 49
4.4.5 Control and Configuration Registers ..... 50
4.4.6 Status Registers ..... 57
4.4.7 COMMAND (Required) ..... 62
4.4.8 Capability Registers ..... 65
4.4.9 MESSAGE_HEADER_INFO (Required) ..... 69
4.4.10 RECEIVE_DETECT (Required) ..... 69
4.4.11 RECEIVE_BUFFER (Required) ..... 70
4.4.12 TRANSMIT (Required) ..... 72
4.4.13 TRANSMIT_BUFFER (Required) ..... 73
4.4.14 VBUS_VOLTAGE (Optional Normative) ..... 74
4.4.15 Voltage Thresholds ..... 75
4.4.16 VENDOR_DEFINED Registers ..... 77
4.5 STANDARD IO SIGNALS ..... 78
4.5.1 STANDARD INPUT SIGNALS (Optional Normative) ..... 78
4.5.2 STANDARD OUTPUT SIGNALS (Optional Normative except Alert#) ..... 78
4.6 TCPC Connection State Diagrams and Flows ..... 80
4.7 PD Communication Operational Model ..... 96
4.7.1 Transmitting an SOP* Message ..... 96
4.7.2 Transmitting a Hard Reset Message ..... 96
4.7.3 Receiving an SOP* message ..... 96
4.7.4 Receiving a Hard Reset message ..... 97
4.7.5 Receiving a Cable Reset message ..... 97
4.8 Power Management ..... 98
4.8.1 I2C Interface ..... 98
4.8.2 PD Messaging ..... 99
4.8.3 CC Status Reporting ..... 99
4.8.4 VBUS Reporting ..... 99
4.8.5 Fault Status Reporting ..... 100
4.9 TCPC Timing Constraints ..... 101
4.10 I2C Physical Interface Specifications ..... 101
A Informative TCPM State Diagrams ..... 104
~~Specification Work Group Chairs / Specification Editors ..... 7~~
~~Specification Work Group Contributors ..... 7~~
~~Revision History ..... 9~~
~~1 Introduction ..... 10~~
~~1.1 Purpose ..... 10~~
~~1.2 Scope ..... 10~~
~~1.3 Related Documents ..... 11~~
~~1.4 Conventions ..... 11~~

**Formatted:** Default Paragraph Font, Check spelling and grammar

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

1.4.1 Precedence ..... 11
1.4.2 Keywords ..... 11
1.4.3 Numbering ..... 12
1.5 Terms and Abbreviations ..... 12
2 Overview ..... 13
2.1 Introduction ..... 13
2.2 USB Type-C Port Controller (TCPC) Interface ..... 13
3 Type-C Port Controller Requirements ..... 14
3.1 Port Power Control for VBUS and VCONN ..... 14
3.2 USB CC Logic ..... 14
3.3 USB-PD Message Delivery ..... 14
3.4 Debug Accessory Detection ..... 15
3.5 TCPC State-Machines ..... 16
3.6 USB Type-C Port Controller Requirements for Source, Sink, and DRP ..... 18
3.6.1 Source Requirements ..... 18
3.6.2 Sink Requirements: ..... 19
3.6.3 Sink with Accessory Support ..... 20
3.6.4 DRP Requirements ..... 21
4 USB Type-C Port Controller Interface ..... 22
4.1 Register Map ..... 23
4.2 TCPC SMBus Optional Normative Requirements ..... 25
4.3 Writing and Reading TCPC Registers ..... 25
4.3.1 Writing Single Byte Registers ..... 25
4.3.2 Reading Single Byte Registers ..... 26
4.3.3 Writing Multiple-Byte Registers ..... 26
4.3.4 Reading Multiple-Byte Registers ..... 27
4.3.5 Writing the TRANSMIT_BUFFER ..... 27
4.3.6 Reading the RECEIVE_BUFFER ..... 28
4.4 Register Definition ..... 29
4.4.1 Identification Registers ..... 29
4.4.2 ALERT Register (Required) ..... 31
4.4.3 Mask Registers ..... 33
4.4.4 CONFIGURE STANDARD OUTPUT (Optional Normative) ..... 36
4.4.5 Control and Configuration Registers ..... 37
4.4.6 Status Registers ..... 43
4.4.7 COMMAND (Required) ..... 49
4.4.8 Capability Registers ..... 52
4.4.9 MESSAGE_HEADER_INFO (Required) ..... 56
4.4.10 RECEIVE_DETECT (Required) ..... 56
4.4.11 RECEIVE_BUFFER (Required) ..... 57
4.4.12 TRANSMIT (Required) ..... 59
4.4.13 TRANSMIT_BUFFER (Required) ..... 60
4.4.14 VBUS_VOLTAGE (Optional Normative) ..... 61

Formatted ...
Formatted ...
Formatted ...

4.4.15 Voltage Thresholds ..... 62
4.4.16 VENDOR_DEFINED Registers ..... 64
4.5 STANDARD IO SIGNALS ..... 65
4.5.1 STANDARD INPUT SIGNALS (Optional Normative) ..... 65
4.5.2 STANDARD OUTPUT SIGNALS (Optional Normative except Alert#) ..... 65
4.6 TCPC Connection State Machine and Flows ..... 67
4.7 PD Communication Operational Model ..... 73
4.7.1 Transmitting an SOP* Message ..... 73
4.7.2 Transmitting a Hard Reset Message ..... 73
4.7.3 Receiving an SOP* message ..... 73
4.7.4 Receiving a Hard Reset message ..... 74
4.8 Power Management ..... 75
4.8.1 I2C Interface ..... 75
4.8.2 PD Messaging ..... 75
4.8.3 CC Status Reporting ..... 75
4.8.4 Vbus Reporting ..... 76
4.8.5 Fault Status Reporting ..... 76
4.9 TCPC Timing Constraints ..... 78
4.10 I2C Physical Interface Specifications ..... 78
A Informative TCPM State-Machine Diagrams ..... 81

FIGURES

Figure 1-1. USB Type-C Port Manager to USB Type-C Port Controller Interface ..... 13
Figure 2-1. TCPC Interface ..... 16
Figure 3-1. Outline of States ..... 19
Figure 3-2. Reference to states ..... 19
Figure 3-3. Rx State Diagram Implemented in TCPC ..... 22
Figure 3-4. Tx State Diagram Implemented in TCPC ..... 23
Figure 3-5. Hard Reset Transmission State Diagram implemented in the TCPC ..... 23
Figure 4-1. Writing Consecutive Registers with or without the SMBUS Protocol ..... 38
Figure 4-2. Reading Consecutive Registers with or without the SMBus Protocol ..... 39
Figure 4-3. Writing a 2-Byte Register with or without the SMBus Protocol ..... 39
Figure 4-4. Reading a 2-Byte Register with or without the SMBus Protocol ..... 40
Figure 4-5. Writing the TRANSMIT_BUFFER with or without the SMBus Protocol ..... 40
Figure 4-6. Reading the RECEIVE_BUFFER with or without the SMBus Protocol ..... 41
Figure 4-7. Automatic VBUS Sink Discharge by the TCPC after a Disconnect ..... 56
Figure 4-8. Transition from vSafe5V to High Voltage ..... 64
Figure 4-9. Transition from High Voltage to vSafe5V ..... 64
Figure 4-10. TCPC Power-On State Diagram ..... 81
Figure 4-11. TCPC State Diagram before a Connection ..... 83
Figure 4-12. TCPC State Diagram after a Connection ..... 83
Figure 4-13. TCPC Debug Accessory State Diagram ..... 86
Figure 4-14. TCPC Sink Debug Accessory Orientation State Diagram ..... 87
Figure 4-15 TCPC Source Debug Accessory Orientation State Diagram ..... 88
Figure 4-16. DRP Initialization and Connection Detection ..... 91
Figure 4-17. Source Disconnect ..... 93
Figure 4-18. Sink Disconnect ..... 95

Copyright © 2015-2016 USB 3.0 Promoter Group. All rights reserved.

Formatted ...

~~Figure 1-1. USB Type-C Port Manager to USB Type-C Port Controller Interface~~ ~~10~~
~~Figure 2-1. TCPC Interface~~ ~~13~~
~~Figure 3-1. Rx State-Machine Implemented in TCPC~~ ~~16~~
~~Figure 3-2. Tx State-Machine Implemented in TCPC~~ ~~17~~
~~Figure 3-3. Hard Reset Transmission State-Machine implemented in the TCPC~~ ~~17~~
~~Figure 4-1. Writing Consecutive Registers with or without the SMBUS Protocol~~ ~~25~~
~~Figure 4-2. Reading Consecutive Registers with or without the SMBus Protocol~~ ~~26~~
~~Figure 4-3. Writing a 2-Byte Register with or without the SMBus Protocol~~ ~~26~~
~~Figure 4-4. Reading a 2-Byte Register with or without the SMBus Protocol~~ ~~27~~
~~Figure 4-5. Writing the TRANSMIT_BUFFER with or without the SMBus Protocol~~ ~~27~~
~~Figure 4-6. Reading the RECEIVE_BUFFER with or without the SMBus Protocol~~ ~~28~~
~~Figure 4-7. Automatic Vbus Sink Discharge by the TCPC after a Disconnect~~ ~~43~~
~~Figure 4-8. Transition from vSafe5V to High Voltage~~ ~~51~~
~~Figure 4-9. Transition from High Voltage to vSafe5V~~ ~~51~~
~~Figure 4-10. TCPC Power-On State Machine~~ ~~67~~
~~Figure 4-11. TCPC State Machine before a Connection~~ ~~68~~
~~Figure 4-12. TCPC State Machine After a Connection~~ ~~69~~
~~Figure 4-13. TCPC State Machine Debug Accessory~~ ~~69~~
~~Figure 4-14. DRP Initialization and Connection Detection~~ ~~70~~
~~Figure 4-15. Source Disconnect~~ ~~71~~
~~Figure 4-16. Sink Disconnect~~ ~~72~~
~~Figure A-1. Rx State Machine Implemented in TCPM~~ ~~81~~
~~Figure A-2. Tx State Machine Implemented in TCPM~~ ~~81~~
~~Figure A-3. Hard Reset State Machine Implemented in TCPM~~ ~~82~~
Figure 4-19. Rx State Diagram Implemented in TCPM ..... 104
Figure 4-20. Tx State Diagram Implemented in TCPM ..... 105
Figure 4-21. Hard Reset State Diagram Implemented in TCPM ..... 107

TABLES

Table 3-1. Required Device_Capability_1 Support Based on Power Role ..... 26
Table 3-2. Required Device_Capability_2 Support Based on Power Role ..... 28
Table 4-1. Register Map ..... 35
Table 4-2. VENDOR_ID Register Definition ..... 42
Table 4-3. PRODUCT_ID Register Definition ..... 42
Table 4-4. DEVICE_ID Register Definition ..... 42
Table 4-5. USBTYPEC_REV Register Definition (Required) ..... 42
Table 4-6. USBPD_REV_VER Register Description ..... 43
Table 4-7. PD_INTERFACE_REV Register Description ..... 43
Table 4-8. ALERT Register Definition ..... 44
Table 4-9. ALERT_MASK Register Definition ..... 46
Table 4-10. POWER_STATUS_MASK Register Definition ..... 47
Table 4-11. FAULT_STATUS_MASK Register Definition ..... 48
Table 4-12. CONFIG_STANDARD_OUTPUT Register Definition ..... 49
Table 4-13. TCPC_Control Register Definition ..... 50
Table 4-14. ROLE_CONTROL Register Definition ..... 52
Table 4-15. Power on Default Conditions ..... 52
Table 4-16. FAULT_CONTROL Register Definition ..... 53
Table 4-17. POWER_CONTROL Register Definition ..... 54
Table 4-18. Discharge Timing Parameters ..... 55
Table 4-19. Debounce requirements ..... 58
Table 4-20. CC_STATUS Register Definition ..... 58
Table 4-21. POWER_STATUS Register Definition ..... 59
Table 4-22. FAULT_STATUS Register Definition ..... 60

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar, Not Small caps
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted** ...
**Formatted** ...
**Formatted** ...

Table 4-23. COMMAND Register Definition ..... 63
Table 4-24. DEVICE_CAPABILITIES_1 Register Definition ..... 65
Table 4-25. DEVICE_CAPABILITIES_2 Register Definition ..... 66
Table 4-26. STANDARD_INPUT_CAPABILITIES Register Definition ..... 67
Table 4-27. STANDARD_OUTPUT_CAPABILITIES Register Definition ..... 68
Table 4-28. MESSAGE_HEADER_INFO Register Definition ..... 69
Table 4-29. RECEIVE_DETECT Register Definition ..... 69
Table 4-30. RECEIVE_BYTE_COUNT Definition ..... 70
Table 4-31. RX_BUF_FRAME_TYPE Definition ..... 70
Table 4-32. RX_BUF_HEADER Definition ..... 71
Table 4-33. RX_BUFFER_DATA_OBJECTS Definition ..... 71
Table 4-34. TRANSMIT Register Definition ..... 72
Table 4-35. TRANSMIT_BYTE_COUNT Definition ..... 73
Table 4-36. TX_BUF_HEADER Definition ..... 73
Table 4-37. TX_BUFFER_DATA_OBJECTS Definition ..... 73
Table 4-38. VBUS_VOLTAGE Register Definition ..... 74
Table 4-39. VBUS_SINK_DISCONNECT_THRESHOLD Register Description ..... 75
Table 4-40. VBUS_STOP_DISCHARGE_THRESHOLD Register Description ..... 75
Table 4-41. VBUS_VOLTAGE_ALARM_HI_CFG Register Description ..... 76
Table 4-42. VBUS_VOLTAGE_ALARM_LO_CFG Register Description ..... 76
Table 4-43. Standard Input Signals ..... 78
Table 4-44. Standard Output Signals ..... 78
Table 4-45. TCPC Timing Constraints ..... 101
Table 4-46. I2C Static Characteristics ..... 102
Table 4-47. I2C Dynamic Characteristics ..... 102
~~Table 4-1. Register Map ..... 23~~
~~Table 4-2. VENDOR_ID Register Definition ..... 29~~
~~Table 4-3. PRODUCT_ID Register Definition ..... 29~~
~~Table 4-4. DEVICE_ID Register Definition ..... 29~~
~~Table 4-5. USBTYPEC_REV Register Definition (Required) ..... 29~~
~~Table 4-6. USBPD_REV_VER Register Description ..... 30~~
~~Table 4-7. PD_INTERFACE_REV Register Description ..... 30~~
~~Table 4-8. ALERT Register Definition ..... 31~~
~~Table 4-9. ALERT_MASK Register Definition ..... 33~~
~~Table 4-10. POWER_STATUS_MASK Register Definition ..... 34~~
~~Table 4-11. FAULT_STATUS_MASK Register Definition ..... 35~~
~~Table 4-12. CONFIG_STANDARD_OUTPUT Register Definition ..... 36~~
~~Table 4-13. TCPC_Control Register Definition ..... 37~~
~~Table 4-14. ROLE_CONTROL Register Definition ..... 38~~
~~Table 4-15. Power on Default Conditions ..... 39~~
~~Table 4-16. FAULT_CONTROL Register Definition ..... 39~~
~~Table 4-17. POWER_CONTROL Register Definition ..... 40~~
~~Table 4-18. Discharge Timing Parameters ..... 41~~
~~Table 4-19. Debounce requirements ..... 44~~
~~Table 4-20. CC_STATUS Register Definition ..... 44~~
~~Table 4-21. POWER_STATUS Register Definition ..... 46~~
~~Table 4-22. FAULT_STATUS Register Definition ..... 47~~
~~Table 4-23. COMMAND Register Definition ..... 50~~
~~Table 4-24. DEVICE_CAPABILITIES_1 Register Definition ..... 52~~
~~Table 4-25. DEVICE_CAPABILITIES_2 Register Definition ..... 53~~
~~Table 4-26. STANDARD_INPUT_CAPABILITIES Register Definition ..... 54~~
~~Table 4-27. STANDARD_OUTPUT_CAPABILITIES Register Definition ..... 54~~
~~Table 4-28. MESSAGE_HEADER_INFO Register Definition ..... 56~~
~~Table 4-29. RECEIVE_DETECT Register Definition ..... 56~~
~~Table 4-30. RECEIVE_BYTE_COUNT Definition ..... 57~~

**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted:** Default Paragraph Font, Check spelling and grammar
**Formatted** ...
**Formatted** ...
**Formatted** ...
**Formatted** ...
**Formatted** ...
**Formatted** ...
**Formatted** ...
**Formatted** ...
**Formatted** ...
**Formatted** ...
**Formatted** ...
**Formatted** ...
**Formatted** ...

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

~~Table 4-31. RX_BUF_FRAME_TYPE Definition ..... 57~~
~~Table 4-32. RX_BUF_HEADER Definition ..... 58~~
~~Table 4-33. RX_BUFFER_DATA_OBJECTS Definition ..... 58~~
~~Table 4-34. TRANSMIT Register Definition ..... 59~~
~~Table 4-35. TRANSMIT_BYTE_COUNT Definition ..... 60~~
~~Table 4-36. TX_BUF_HEADER Definition ..... 60~~
~~Table 4-37. TX_BUFFER_DATA_OBJECTS Definition ..... 60~~
~~Table 4-38. VBUS_VOLTAGE Register Definition ..... 61~~
~~Table 4-39. VBUS_SINK_DISCONNECT_THRESHOLD Register Description ..... 62~~
~~Table 4-40. VBUS_STOP_DISCHARGE_THRESHOLD Register Description ..... 62~~
~~Table 4-41. VBUS_VOLTAGE_ALARM_HI_CFG Register Description ..... 63~~
~~Table 4-42. VBUS_VOLTAGE_ALARM_LO_CFG Register Description ..... 63~~
~~Table 4-43. Standard Input Signals ..... 65~~
~~Table 4-44. Standard Output Signals ..... 65~~
~~Table 4-45. TCPC Timing Constraints ..... 78~~
~~Table 4-46. I2C Static Characteristics ..... 79~~
~~Table 4-47. I2C Dynamic Characteristics ..... 79~~

**Formatted:** Default Paragraph Font, Check spelling and grammar

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

## Specification Work Group Chairs / Specification Editors

| | |
|---|---|
| Intel Corporation (USB 3.0 Promoter company) | Christine Krause – Subgroup WG co-chair, Specification Co-author<br>Chee Lim Nge – Subgroup WG co-chair, Specification Co-author |

## Specification Work Group Contributors

| | | | |
|---|---|---|---|
| Advanced Micro Devices | Will Harris<br>Jason Hawken | Joseph Scanlon<br>Peter Teng | Ken Xue |
| Analogix Semiconductor, Inc. | Greg Stewart | | |
| Apple | Sree Anantharaman<br>William Ferry<br>Arulchandran Paramasivam | Reese Schreiber<br>David Sekowski<br>Sascha Tietz | Hsiao-Ping Jennifer Tsai |
| Cadence Design Systems, Inc. | Jacek Duda<br>Pawel Eichler | Wojciech Kloska | Michal Staworko |
| Canova Tech | Matteo Casalin | Davide Ghedin | Nicola Scantamburlo |
| Cypress Semiconductor | Anup Nayak<br>Jagadeesan Raj | Subu Sankaran<br>Ganesh Subramaniam | |
| Dell Inc. | Mohammed hijazi<br>Marcin Nowak | Siddhartha Reddy<br>Merle Wood | |
| DisplayLink (UK) Ltd. | Pete Burgers<br>Dan Ellis | Kevin Jacobs<br>Richard Petrie | Jason Young |
| Ellisys | Mario Pasquali | Chuck Trefts | |
| Etron Technology, Inc. | Shihmin Hsu | Chien-Cheng Kuo | |
| Fairchild Semiconductor | Oscar Freitas | Christian Klein | Erik Maier |
| Fresco Logic Inc. | Tim Barilovits | Bob McVay | Jeffrey Yang |
| Google Inc. | Alec Berg<br>Jim Guerin<br>Mark Hayter | Sameer Nanda<br>Vincent Palatin<br>David Schneider | Scott Collyer |
| Granite River Labs | Mike Engbretson | | |
| Hewlett Packard | Roger Benson | Robin Castell | |
| Intel Corporation | Bob Dunstan<br>Abdul Ismail<br>Sanjeev Jahagirdar | Vijaykumar Kadgi<br>Henrik Leegaard<br>Tim McKee | Brad Saunders<br>Karthi Vadivelu |
| Keysight Technologies Inc. | Jit Lim | | |
| Lattice Semiconductor Corp | Young Il Kim | Thomas Watzka | |
| MCCI Corporation | Terry Moore | Sherri Russo | Chris Yokum |
| Microchip Technology Inc. | Josh Averyt<br>Mark Bohm<br>Shannon Cash | Brian Marley<br>Santosh shettty<br>John Sisto | Richard Wahler<br>Neil Winchester |
| Microsoft Corporation | Randy Aull<br>Anthony Chen<br>Vivek Gupta<br>David Hargrove | Robbie Harris<br>Teemu Helenius<br>Kai Inha<br>Jayson Kastens | Ismo Manninen<br>Rahul Ramadas<br>Nathan Sherman<br>Tatu Tomppo |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

| | | | |
|---|---|---|---|
| MQP Electronics Ltd. | Sten Carlsen | Pat Crowe | |
| NXP Semiconductors | Ken Jaramillo<br>Abhijeet Kulkarni | Vijendra Kuroodi<br>Krishnan TN | Bart Vertenten |
| Nuvoton | Victor Flachs | Nimrod Peled | |
| ON Semiconductor | Bryan McCoy | | |
| Parade Technologies, Inc. | Jian Chen<br>Craig Wiley | Paul Xu<br>Alan Yuen | |
| Qualcomm, Inc | Craig Aiken | Shadi Hawawini | |
| Realtek Semiconductor Corp. | Charlie Hsu<br>Ray Lee | Ryan Lin<br>Terry Lin | Changhung Wu |
| Renesas Electronics Corp. | Kiichi Muto | Hajime Nozaki | |
| Richtek Technology Corporation | Kenny Chan<br>HM Chang<br>Patrick Change<br>Bryan Huang | Spice Huang<br>Chunan Kuo<br>Tony Lai<br>Heinz Wei | Scott Wu<br>Alex Yang<br>Ming-Shih Yu |
| Ricoh Company Ltd. | Yasuyuki Hayashi<br>Tatsuya Irisawa | Satoshi Oie<br>Yuuji Tsutsui | |
| ROHM Co., Ltd. | Kris Bahar<br>Ruben Balbuena | Nobutaka Itakura<br>Yoshinori Ohwaki | Takashi Sato |
| STMicroelectronics | Jérôme Bach<br>Nathalie Ballot<br>Christophe lorin | Meriem Mersel<br>Federico Musarra<br>Richard O'Connor | Legrand Pascal |
| Synopsys, Inc. | Zongyao Wen | | |
| Texas Instruments | Felipe Balbi | Scott Jackson | Deric Waters |
| VIA Technologies, Inc. | Jay Tseng | Fong-Jim Wang | |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

## Revision, Version History

Formatted: PARAGRAPH

Formatted Table

| Revision | ~~Revision~~Version | Date | Description |
|---|---|---|---|
| 1.0 | 1.1 | July 2016 | Update Release<br>Reprint release including incorporation of all approved ECNs as of the revision date plus editorial clean-up. |
| 1.0 | 1.0 | October 15, 2015 | Initial Release |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

# 1 Introduction

With the continued success of USB Power Delivery, there exists a need to define a common interface from a USB Type-C™ Port Manager to a Simple USB Type-C Port Controller. This specification defines this interface.

Figure 1-1 shows the interconnection between the USB Type-C Port Manager, TCPM, and three USB Type-C Port Controllers, TCPCs. One TCPM may be used to drive multiple TCPCs subject to the timing constraints defined in the ***USB PD~~USB PD~~*** Specification. The connection between the TCPM and the TCPC is defined as the USB Type-C Port Controller Interface, TCPCI.

Formatted: bookmark Char, Font: Not Bold

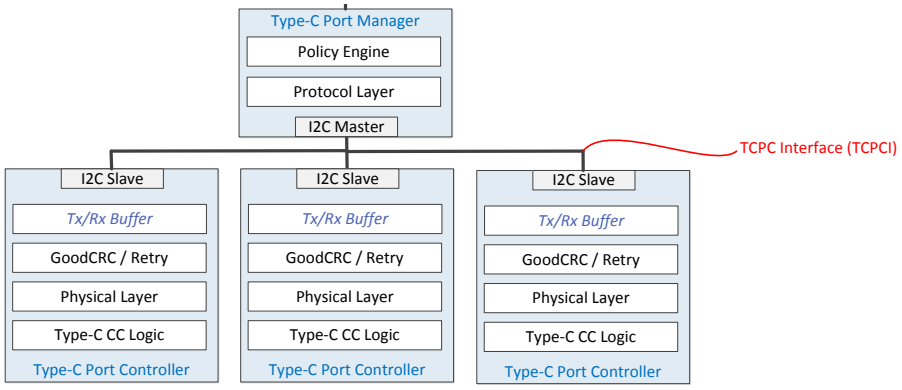



**Figure 1-~~1~~1~~-1~~. USB Type-C Port Manager to USB Type-C Port Controller Interface**

## 1.1 Purpose

The USB Type-C Port Controller Interface, TCPCI, is the interface between a USB Type-C Port Manager and a USB Type-C Port Controller. This specification standardizes the communication between the USB Type-C Port Manager (TCPM) and the USB Type-C Port Controller (TCPC) while meeting the ***USB Type-C~~USB Type-C~~*** Power Delivery requirements.

Formatted: bookmark Char, Font: Not Bold

The goal of the USB Type-C Port Controller Interface (TCPCI) is to provide a defined interface between a TCPC and a TCPM in order to standardize and simplify USB Type-C Port Manager implementations.

The TCPC is a functional block which encapsulates VBUS and VCONN power controls, ***USB Type-C~~USB Type-C~~*** CC logic, and the ***USB PD~~USB PD~~*** BMC physical layer and protocol layer other than the message creation.

Formatted: bookmark Char, Font: Not Bold

Formatted: bookmark Char, Font: Not Bold

## 1.2 Scope

This specification is intended as a supplement to ***USB 3.1~~USB 3.1~~***, ***USB Type-C~~USB Type-C~~***, and ***USB PD~~USB PD~~*** specifications. It addresses only the elements required to implement and support the ***USB Type-C~~USB Type-C~~*** Port Controller.

Formatted: bookmark Char, Font: Not Bold

Formatted: bookmark Char, Font: Not Bold

Formatted: bookmark Char, Font: Not Bold

Formatted: bookmark Char, Font: Not Bold

Normative information is provided to allow interoperability of components designed to this specification. Informative information, when provided, may illustrate possible design implementations.

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

### 1.3 Related Documents

| | |
|---|---|
| **USB 3.1** | *Universal Serial Bus Revision 3.1 Specification*<br>This includes the entire document release package.<br>http://www.usb.org/developers/docs |
| **USB PD** | *USB Power Delivery Specification, Revision 2.0, V1.2 March 25~~August 11~~, ~~2014~~2016*<br>http://www.usb.org/developers/docs |
| **USB Type-C** | *USB Type-C Connector Specification, Revision 1.~~0~~2, ~~August~~ March ~~11~~25, ~~2014~~2016*<br>http://www.usb.org/developers/docs |

### 1.4 Conventions

#### 1.4.1 Precedence

If there is a conflict between text, figures, and tables, the precedence shall be tables, figures, and then text.

#### 1.4.2 Keywords

The following keywords differentiate between the levels of requirements and options.

##### 1.4.2.1 Informative

Informative is a keyword that describes information within this specification that intends to discuss and clarify requirements and features as opposed to mandating them.

##### 1.4.2.2 May

May is a keyword that indicates a choice with no implied preference.

##### 1.4.2.3 N/A

N/A is a keyword that indicates a field or value is not applicable and has no defined value and shall not be checked or used by the recipient.

##### 1.4.2.4 Normative

Normative is a keyword that describes features mandated by this specification.

##### 1.4.2.5 Optional

Optional is a keyword that describes features not mandated by this specification. However, if an optional feature is implemented, the feature shall be implemented as defined by this specification (optional normative).

##### 1.4.2.6 Reserved

Reserved is a keyword indicating reserved bits, bytes, words, fields, and code values that are set-aside for future standardization. Their use and interpretation may be specified by future extensions to this specification and, unless otherwise stated, shall not be utilized or adapted by vendor implementation. A reserved bit, byte, word, or field shall be set to zero by the sender and shall be ignored by the receiver. Reserved field values shall not be sent by the sender, and if received, shall be ignored by the receiver.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

#### 1.4.2.7 Shall

Shall is a keyword indicating a mandatory (normative) requirement. Designers are mandated to implement all such requirements to ensure interoperability with other compliant Devices.

#### 1.4.2.8 Should

Should is a keyword indicating flexibility of choice with a preferred alternative equivalent to the phrase "it is recommended that".

### 1.4.3 Numbering

Numbers immediately followed by a lowercase "b" (e.g., 01b) are binary values. Numbers immediately followed by an uppercase "B" are byte values. Numbers immediately followed by a lowercase "h" (e.g., 3Ah) are hexadecimal values. Numbers not immediately followed by either a "b", "B", or "h" are decimal values.

## 1.5 Terms and Abbreviations

| Term | Description |
|---|---|
| BMC | Biphase Mark Coding |
| LPM | Local Policy Manager |
| LPMI | Local Policy Manager Interface |
| OPM | Operating System Policy Manager |
| PPM | Platform Policy Manger |
| PPMI | Platform Policy Manager Interface |
| TCPC | USB Type-C Port Controller |
| TCPCI | USB Type-C Port Controller Interface |
| TCPM | USB Type-C Port Manager |
| Snk.Rp | Sink CC pin above minimum vRd-Connect, ~~*USB Type-CUSB Type-C*~~ |
| Snk.Open | Sin CC pin below maximum vRa, ~~*USB Type-CUSB Type-C*~~ |
| Src.Ra | Source CC pin above vOPEN, ~~*USB Type-CUSB Type-C*~~ |
| Src.Rd | Source CC pin within the vRd range, ~~*USB Type-CUSB Type-C*~~ |
| Src.Open | Source CC pin below maximum vRa |
| OCP | Over-current Protection |
| OVP | Over-voltage Protection |
| vSafe0V | Safe operating voltage at "zero volts" per ~~*USB PDUSB PD*~~ |
| vSafe5v | Safe operating voltage at 5V per ~~*USB PDUSB PD*~~ |

**Formatted:** bookmark Char, Font: 8 pt, Not Bold

**Formatted:** bookmark Char, Font: 8 pt, Not Bold

**Formatted:** bookmark Char, Font: 8 pt, Not Bold

**Formatted:** bookmark Char, Font: 8 pt, Not Bold

**Formatted:** bookmark Char, Font: 8 pt, Not Bold

**Formatted:** bookmark Char, Font: 8 pt, Not Bold

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

# 2 Overview

## 2.1 Introduction

## 2.2 USB Type-C Port Controller (TCPC) Interface

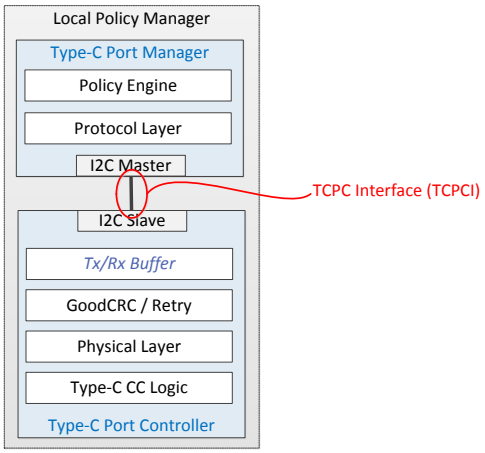



**Figure 2-~~12-1~~. TCPC Interface**

The USB Type-C Port Controller Interface, TCPCI, is the interface between a USB Type-C Port Manager and a USB Type-C Port Controller. The goal of the USB Type-C Port Controller Interface (TCPCI) is to provide a defined interface between a TCPC and a TCPM in order to standardize and simplify USB Type-C Port Manager implementations.

The TCPC is a functional block which encapsulates VBUS and VCONN power controls, ***USB Type-C~~USB Type-C~~*** CC logic, and the ***USB PD~~USB PD~~*** BMC physical layer and protocol layer other than the message creation. The TCPC shall NOT include support for USB PD BFSK.

**Formatted:** bookmark Char, Font: Not Bold

**Formatted:** bookmark Char, Font: Not Bold

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

## 3 Type-C Port Controller Requirements

This chapter describes the requirements of a USB Type-C Port Controller. The TCPC has three functions:

- ***~~USB Type-C~~USB Type-C*** Port Power Control for VBUS and VCONN (required)
- ***~~USB Type-C~~USB Type-C*** CC Control and sensing (required)
- ***USB PD~~USB PD~~*** Message delivery (required)
- Standard Inputs and Outputs are defined for simplified external interfacing (optional)

The TCPC uses I2C to communicate with the TCPM. The TCPC is an I2C slave with Alert# signal for requesting attention.

### 3.1 Port Power Control for VBUS and VCONN

A Source capable TCPC shall provide ~~a~~ the register~~s~~ required ~~which~~ to allow~~s~~ the TCPM to control VBUS Sourcing. A Sink capable TCPC shall provide ~~a~~ the register~~s~~ ~~which~~ to allow~~s~~ the TCPM to Control VBUS Sinking.

To ensure safety in case the I2C interface fails, ~~the~~ a TCPC ~~that is~~ sourcing VBUS higher than 5V shall autonomously stop sourcing VBUS if the Sink is detached.

The TCPC shall implement a force discharge circuit if it supports sourcing VBUS. A low current bleed discharge may ~~also~~ be implemented to discharge VBUS. ~~The~~ F~~f~~orce discharge is a larger current discharge used to discharge VBUS to below vSafe0V upon detecting a Disconnect per ***~~USB Type-C~~USB Type-C*** (exiting the Attached.SRC state). ~~The discharge shall be available for both Source and Sink.~~

A TCPC shall include monitoring for the presence of VBUS (VSafe5V, VSafe0V). The TCPC shall implement high and low voltage alarms if it Sinks or Sources voltage greater than VSafe5V.

A Source or DRP TCPC shall include control for VCONN sourcing. A Sink TCPC shall include control for VCONN sourcing if VCONN Swap or Sink w/Accessory is supported. VCONN sourcing shall meet the tVCONNON and tVCONNOFF timing requirement per ***~~USB Type-C~~USB Type-C***.

A TCPC shall implement low power states as defined in this specification.

### 3.2 USB CC Logic

The TCPC shall implement logic for controlling the CC pins on the USB Type-C Connector. The TCPC shall implement a method to control the Port Power Role and to report the state of the CC lines, Rp/Rd control, and CC sense/debounce/interrupt.

### 3.3 USB-PD Message Delivery

The TCPC shall implement BMC encoding. The TCPC shall NOT include support for ***USB PD~~USB PD~~*** BFSK. The TCPC shall implement the portion of the Protocol layer in the ***USB PD~~USB PD~~*** specification as shown in Figure 3-2, and 3-3. The TCPC is opaque from a ***USB PD~~USB PD~~*** point of view. The TCPC sends and receives messages constructed in the TCPM and places them on the CC connections. The TCPC does not interpret the ***USB PD~~USB PD~~*** messages.

The TCPC shall implement the entire ***USB PD~~USB PD~~*** PHY layer with BMC encoding. The TCPC shall implement a portion of the Protocol Layer Transmit state ~~machine~~diagram.

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

**Formatted:** bookmark Char, Font: Not Bold
**Formatted:** bookmark Char, Font: Not Bold
**Formatted:** bookmark Char, Font: Not Bold
**Formatted:** Font color: Auto
**Formatted:** Font color: Auto
**Formatted:** Font color: Auto
**Formatted:** bookmark Char, Font: Not Bold
**Formatted:** bookmark Char, Font: Not Bold
**Formatted:** bookmark Char, Font: Not Bold
**Formatted:** bookmark Char, Font: Not Bold
**Formatted:** bookmark Char, Font: Not Bold
**Formatted:** bookmark Char, Font: Not Bold
**Formatted:** bookmark Char, Font: Not Bold
**Formatted:** bookmark Char, Font: Not Bold
**Formatted:** bookmark Char, Font: Not Bold

- CRCReceiveTimer (PRL_Tx_wait_for_Phy_Response_state)
- RetryCounter (PRL_Tx_Check_RetryCounter State)
- MessageID is not checked in the TCPC when a non-GoodCRC message is received. Retried messages that are received are passed to the TCPM via I2C
- Received GoodCRC must match the transmitted MessageID and SOP type before it is considered valid
- Two things allow the TCPM to track the MessageID even when asynchronous messages are received
    - If ALERT.ReceiveSOP*MessageStatus is not cleared when the TCPM requests a TRANSMIT then the I2C command is NAK'd or TransmitSOP*MessageDiscarded bit in the ALERT register is asserted.
    - If a message is received before the TCPC has processed a transmit request, it asserts the TransmitSOP*MessageDiscarded bit in the ALERT register.
- BIST handling shall be as follows: Each incoming BIST message may be passed up to the policy engine as is any other incoming ***USB PD~~USB PD~~*** Message, or responded to with a GoodCRC without passing to the policy engine. The TCPC shall provide a mechanism to allow the policy engine to send a BIST Continuous Carrier Mode 2 message for tBistContMode.

Formatted: bookmark Char, Font: Not Bold

### 3.4 Debug Accessory Detection

The TCPC may implement autonomous detection of the Debug Accessory State (vRd/vRd) per ***USB Type-C~~USB Type-C~~***. This allows the TCPC to indicate a vRd/vRd connection without TCPM involvement, and indicates this via the DebugAccessoryConnected# output and POWER_STATUS.DebugAccessoryConnected. The TCPC performs autonomous detection of the Debug Accessory state if TCPC_CONTROL.DebugAccessoryControl=0b.

Formatted: bookmark Char, Font: Not Bold

The TCPM may control entry to the Debug Accessory Detected state by setting TCPC_CONTROL.DebugAccessoryControl=1b.

The behavior in the Debug Accessory state is defined in ***USB Type-C~~USB Type-C~~*** in Appendix B.

Formatted: bookmark Char, Font: Not Bold

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

Formatted: Heading 2

## 3.5 State Diagram Introduction

The TCPC state diagrams defined this specification are Normative and shall define the operation of the TCPC. Note that these state diagrams are not intended to replace a well written and robust design. Figure 3-1 shows an outline of the states defined in the following sections. At the top there is the name of the state. This is followed by "Actions on entry" a list of actions carried out on entering the state and in some states "Actions on exit" a list of actions carried out on exiting the state.

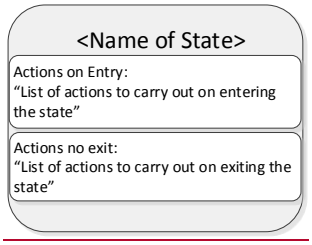



**Figure 3-1. Outline of States**

Transitions from one state to another are indicated by arrows with the conditions listed on the arrow. Where there are multiple conditions these are connected using either a logical OR "|" or a logical AND "&". The inverse of a condition is shown with a "NOT" in front of the condition. In some cases there are transitions which can occur from any state to a particular state. These are indicated by an arrow which is unconnected to a state at one end, but with the other end (the point) connected to the final state. In some state diagrams it is necessary to enter or exit from states in other diagrams. Figure 3-2 indicates how such references are made. The reference is indicated with a hatched box. The box contains the name of the referenced state.

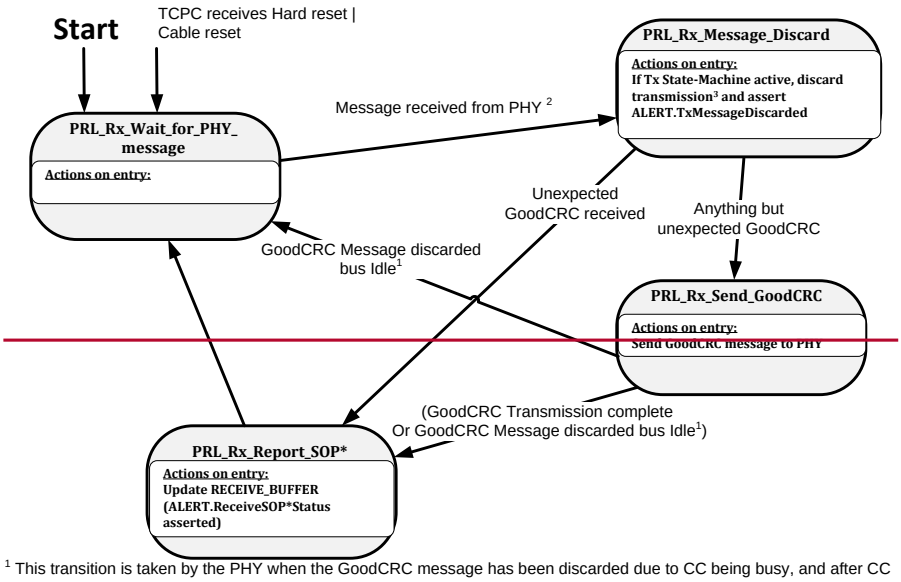
**Figure 3-2. Reference to states**

### ~~3.5~~3.6 TCPC Protocol Layer State~~-Machines~~ Operation

This section describes the normative TCPC Protocol layer ~~State-Machines for the TCPC~~state operation. The informative TCPM Protocol Layer ~~State-Machines~~state operation can be found in Appendix A.

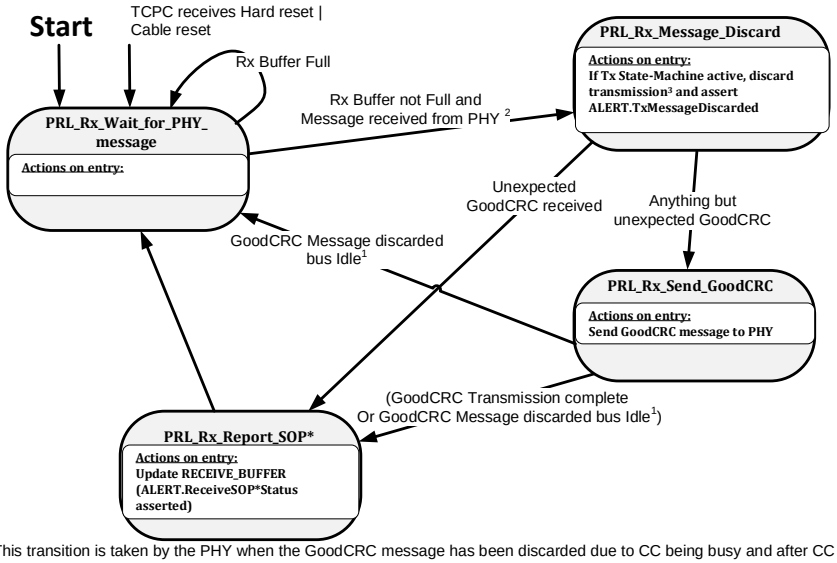
Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.




[1] This transition is taken by the PHY when the GoodCRC message has been discarded due to CC being busy, and after CC becomes idle again (see USB-PD specification). Two alternate allowable transitions are shown.
[2] Messages do not include Hard Reset or Cable Reset signals or expected GoodCRC messages (GoodCRC messages are only expected after the TCPC PHY has received the tx message and the TCPC Tx state-machine is in the PRL_Tx_Wait_for_PHY_response state).
[3] The TCPC may not discard the transmission if the received message is a Ping message.

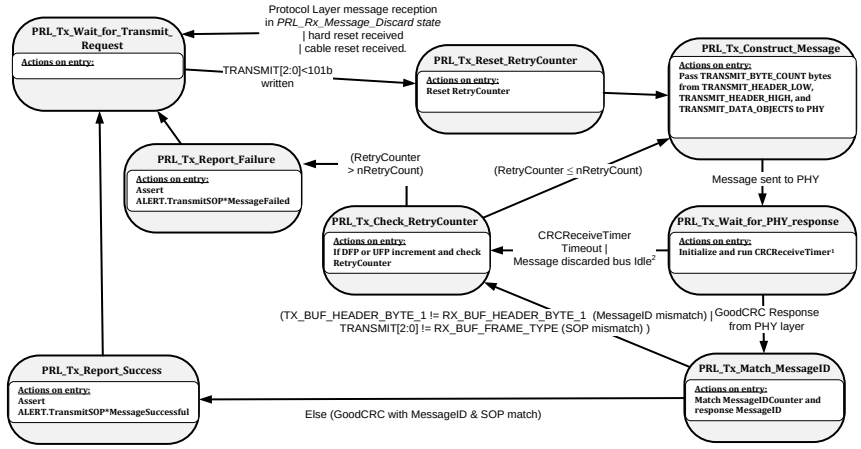
Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.




[1] This transition is taken by the PHY when the GoodCRC message has been discarded due to CC being busy and after CC becomes idle again (see USB-PD specification). Two alternate allowable transitions are shown.
[2] Messages do not include Hard Reset or Cable Reset signals or expected GoodCRC messages (GoodCRC messages are only expected after the TCPC PHY has received the tx message and the TCPC Tx state-machine is in the PRL_Tx_Wait_for_PHY_response state).
[3] The TCPC may not discard the transmission if the received message is a Ping message.

**Figure 3-~~3~~3-~~1~~1. Rx State-~~Machine~~ Diagram Implemented in TCPC**

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.




[1] The CRCReceiveTimer is only started after the PHY has sent the message. If the message is not sent due to a busy channel then the CRCReceiveTimer will not be started (see USB-PD Rev2.0 v1.1 Section 6.5.1).
[2] This indication is sent by the PHY Layer when a message has been discarded due to $V_{BUS}$ or CC being busy, and after $V_{BUS}$ or CC becomes idle again (see USB-PD REv2.0 v1.1 Section 5.8). The CRCReceiveTimer is not running in this case since no message has been sent.

**Figure 3-~~43-2~~. Tx State ~~Machine~~ Diagram Implemented in TCPC**

Formatted: Keep with next

Formatted: Normal

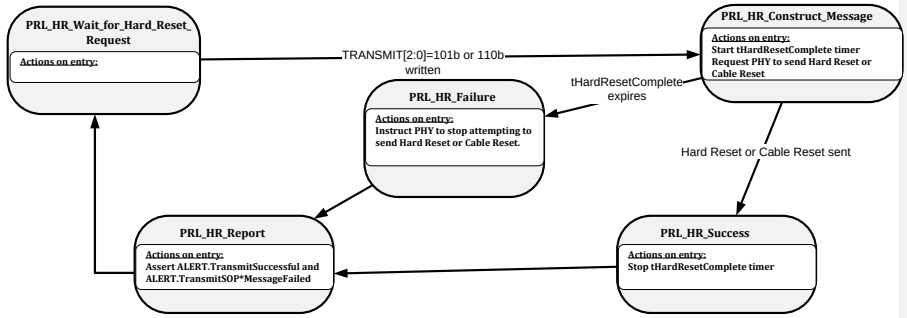



**Figure 3-~~53-3~~. Hard Reset Transmission State Diagram ~~Machine~~ implemented in the TCPC**

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### 3.7 USB Type-C Port Controller Requirements for Source, Sink, and DRP

A TCPC shall implement the DEVICE_CAPABILITES_1 and DEVICE_CAPABILITES_2 registers as defined in Section 4.4.8.1. A TCPC shall support the DEVICE_CAPABILITIES_1 register for the applicable Power Role as defined in Table 3-1. A TCPC shall implement the DEVICE_CAPABILITIES_2 register for the applicable Power Role as defined in Table 3-2.

**Formatted:** Left: 1", Right: 1", Top: 1.25", Bottom: 1.25", Width: 11", Height: 8.5"

**Formatted:** PARAGRAPH

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

**Table 3-1. Required Device Capability 1 Support Based on Power Role**

| Power Role | | Device Capabilities 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | B14 OCP | B13 OVP | B12 Bleed Discharge | B11 Force Discharge | B10 VBUS Alarm Meas | B9..8 Rp | B7..5 Roles | B4 SOP'/SOP" DBG | B3 Source VCONN | B2 Sink VBUS | B1 Source HV | B0 Source VBUS |
| Source-only (>5V Vbus) | | O[2] | O[3] | O | R | R | 01b 10b | 001b | O | R | O | R | R |
| Source-only (5V Vbus Default) | | O[2] | O[3] | O | R | O | 00b | 001b | O | R | O | O | R |
| Sink-only (>5V) | | O | O[3] | O | O | R | 00b | 010b | O | O | R | O | O |
| Sink-only (5V Default) | | O | O[3] | O | O | O | 00b | 010b | O | O | R | O | O |
| DRP | Toggling (Source/Sink) (>5V) | O[2] | O[3] | O | R | R | 01b 10b | 100b 101b 110b | O | R | R | R | R |
| | Toggling (Source/Sink) (5V Default) | O[2] | O[3] | O | R | O | 00b | 100b 101b 110b | O | R | R | O | R |
| | Sourcing Device (>5V) | O[2] | O[3] | O | R | R | 01b 10b | 100b 101b 110b | O | R | O | R | R |
| | Sourcing Device (5V Default) | O[2] | O[3] | O | R | O | 00b | 100b 101b 110b | O | R | O | O | R |
| | Sinking Host (>5V) | O | O[3] | O | O | R | 00b | 100b 101b 110b | O | O | R | O | O |
| | Sinking Host (5V Default) | O | O[3] | O | O | O | 00b | 100b 101b 110b | O | O | R | O | O |

Notes:
1. R=Required and O=Optional
2. Required at the platform level per USB-PD. OCP can be integrated into the TCPC or external to the TCPC. This bit indicates the TCPC supports reporting OCP through FAULT_STATUS register. If OCP is external to the TCPC, the OCP shall be connected to the Standard Input Signal, VBUS External Over Current Fault. If this bit is not set, then OCP event is not visible to TCPC.

Formatted: Caption, Keep with next
Formatted: Font color: Auto
Formatted Table
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

~~**3.6**~~ ~~Device_Capabilities_1.VbusOVPReporting (B13) defines if reporting of the OVP event is supported or not. Required now per latest update in USB-PD rev 1.2.~~

**Formatted:** TableFootnoteStepList, Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.43" + Indent at: 0.68"

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

**Table ~~3-~~2. Required Device Capability 2 Support Based on Power Role**

| Power Role | | Device Capabilities 2 | | | | | |
|---|---|---|---|---|---|---|---|
| | | B15..8 | B7 Sink Disconnect Detection | B6 Stop Discharge Threshold | B5..4 VBUS Alarm | B3..1 VCONN Power | B0 VCONN Over Current Fault Capable |
| Source-only (>5V) | | | O | R | O | 000b or other | O |
| Source-only (5V Default) | | | O | O | O | 000b or other | O |
| Sink-only (>5V) | | | R | O | O | Don't care | O |
| Sink-only (5V Default) | | | O | O | O | Don't care | O |
| DRP | Toggling (Source/Sink) (>5V) | | R | R | O | 000b or other | O |
| | Toggling (Source/Sink) (5V Default) | | O | O | O | 000b or other | O |
| | Sourcing Device (>5V) | | O | R | O | 000b or other | O |
| | Sourcing Device (5V Default) | | O | O | O | 000b or other | O |
| | Sinking Host (>5V) | | R | O | O | 000b or other | O |
| | Sinking Host (5V Default) | | O | O | O | 000b or other | O |

Notes:

1. R=Required and O=Optional

Formatted: Caption, Keep with next
Field Code Changed
Field Code Changed
Formatted: Font color: Auto
Formatted Table
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### ~~3.6.1~~3.7.1 Source Requirements

A TCPC, which supports Source port operation, is defined as follow:

1. A Source TCPC shall provide control of VBUS source path (see Table 4-23. COMMAND Register Definition).
2. A Source TCPC may provide over voltage protection and over current protection circuitry for the VBUS source path (see FAULT_STATUS.OCP/OVP and FAULT_CONTROL.OCP/OVP).
3. A Source TCPC shall provide control of a VCONN switch (see POWER_CONTROL.VCONNPowerSupported and POWER_CONTROL.EnableVconn).
4. A Source TCPC may include monitoring for the presence of VCONN (see POWER_STATUS.VCONNPresent).
5. ~~4.~~A Source TCPC shall support Device_Capabilities_1 and Device_Capabilities_2 register for the Source-only (>5V) or Source-only (5V Default) Power Role as defined in Table 3-1 and Table 3-2.

| Name | Functionality |
|---|---|
| **USB-PD** | |
| VCONN Swap | Optional |
| Power Role Swap Support | Optional |
| USB-PD Support | Optional |
| **CC CONTROL** | |
| CC Detect Status | Required |
| Port Disable | Required (Rp to zOpen) |
| Roles Supported | SRC (Rp default, 1.5A, 3A) indicated in DEVICE_CAPABILITIES_1.SourceResistorSupported<br>SNK (Rd) Optional |
| **PORT POWER CONTROL** | |
| Power Status | Required |
| Supply VCONN | Required |
| Sink VBUS | Optional |
| Supply VBUS | Required |
| Dead Battery | Required in DRP (present Rd when no power)<br>Not required for Source only |

Field Code Changed

**Formatted:** Font: 10 pt, Not Bold, Font color: Auto, Do not check spelling or grammar

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

### ~~3.6.2~~3.7.2 Sink Requirements:

A TCPC, which supports Sink port operation, is defined as follow:

1. A Sink TCPC shall contain CC logic that implements a mechanism to present Rd in a dead battery condition (see Table 4-15. Power on Default Conditions).
2. A Sink TCPC may include the monitoring of the presence of VCONN (see POWER_CONTROL.VCONNPowerSupported and POWER_STATUS.VCONNPresent).
3. A Sink TCPC shall provide control of VBUS sink path (see COMMAND).
4. A Sink TCPC shall provide a mechanism for detecting a Disconnect if it is capable of sinking a voltage greater than vSafe5V (see Section 4.4.15.1).
5. A Sink TCPC shall provide a mechanism for detecting vSafe0V.
6. ~~5.~~A Sink TCPC shall support Device_Capabilities_1 and Device_Capabilities_2 register for the Sink-only (>5V) or Sink-only (5V Default) Power Role as defined in Table 3-1 and Table 3-2.

Field Code Changed

**Formatted:** Font: 10 pt, Not Bold, Font color: Auto, Do not check spelling or grammar

| Name | Functionality |
|---|---|
| **USB-PD** | |
| VCONN Swap | Optional |
| Power Role Swap Support | Optional |
| USB-PD Support | Optional |
| **CC CONTROL** | |
| CC Detect Status | Required |
| Port Disable | Required (Rd to zOpen) |
| Roles Supported | SNK (Rd) Required<br>SRC (Rp default, 1.5A, 3A) Optional |
| **PORT POWER CONTROL** | |
| Power Status | Required |
| Supply VCONN | Optional, but required if VCONN Swap supported |
| Sink VBUS | Required |
| Supply VBUS | Optional |
| Dead Battery | Required (present Rd when no power) |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### ~~3.6.3~~3.7.3 Sink with Accessory Support

A TCPC, which supports Sink with Accessory Support operation, is defined as follow:

1. A Sink TCPC shall contain CC logic that implements a mechanism to present Rd in a dead battery condition (see Table 4-15. Power on Default Conditions).
2. A Sink TCPC shall provide control of VCONN source path (see POWER_CONTROL.VCONNPowerSupported and POWER_CONTROL.EnableVCONN).
3. A Sink TCPC may include the monitoring of the presence of VCONN (see POWER_STATUS.VCONNPresent).
4. A Sink TCPC shall provide control of VBUS sink path (see COMMAND).
5. A Sink TCPC shall provide a mechanism for detecting a Disconnect if it is capable of sinking a voltage greater than vSafe5V (see Section 4.4.15.1).
6. A Sink TCPC shall provide a mechanism for detecting vSafe0V.
~~6.~~7. A Sink TCPC shall support Device_Capabilities_1 and Device_Capabilities_2 register for the Sink-only (>5V) or Sink-only (5V Default) Power Role as defined in Table 3-1 and Table 3-2.

Field Code Changed

**Formatted:** Font: 10 pt, Not Bold, Font color: Auto, Do not check spelling or grammar

Sink with Accessory support is optional, but if implemented shall follow the table below.

| Name | Functionality |
|---|---|
| **USB-PD** | |
| VCONN Swap | Required |
| Power Role Swap Support | Optional |
| USB-PD Support | Required |
| **CC CONTROL** | |
| CC Detect Status | Required |
| Port Disable | Required (Rp to zOpen) |
| Roles Supported | SNK (Rd) Required<br>SRC (Rp default) Required |
| **PORT POWER CONTROL** | |
| Power Status | Required |
| Supply VCONN | Required |
| Sink VBUS | Required |
| Supply VBUS | Optional |
| Dead Battery | Required (present Rd when no power) |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### ~~3.6.4~~3.7.4 DRP Requirements

A TCPC, which supports Dual Role Port operation, is defined as follow:

1. A Dual Role TCPC shall contain CC logic to detect the insertion of a Source, Sink, and Audio and debug accessory (see ROLE_CONTROL).
2. A Dual Role TCPC shall contain CC logic that implements a mechanism to present Rd in a dead battery condition (see CC_STATUS).
3. A Dual Role TCPC shall provide control of VBUS source path (see COMMAND).
4. A Dual Role TCPC shall provide control for a Vconn switch (see POWER_CONTROL.VCONNPowerSupported and POWER_CONTROL.EnableVCONN)
5. The TCPC shall include the monitoring of the presence of VCONN (see POWER_STATUS.VCONNPresent).
6. A DRP TCPC shall provide a mechanism for detecting a Disconnect if it is capable of sinking a voltage greater the vSafe5V (see Section 4.4.15.1).
7. A DRP TCPC shall provide a mechanism for detecting vSafe0V.
8. A DRP TCPC shall support Device_Capabilities_1 and Device_Capabilities_2 register for at least the DRP Toggling, Sourcing Device, and Sinking Host at 5V Default Power Roles as defined in Table 3-1 and Table 3-2.

~~7.~~9.

| Name | Functionality |
|---|---|
| **USB-PD** | |
| VCONN Swap | Optional |
| PR Swap Support | Optional |
| USB-PD Support | Optional |
| **CC CONTROL** | |
| CC Detect Status | Required |
| Port Disable | Required (Rp to zOpen) |
| Roles Supported | SRC (Rp default, 1.5A, 3A) indicated in DEVICE_CAPABILITIES_1.SourceResistorSupported<br>SNK (Rd) Required |
| **PORT POWER CONTROL** | |
| Power Status | Required |
| Supply VCONN | Required |
| Sink VBUS | Required if Dead Battery Supported |
| Supply VBUS | Required |
| Dead Battery | Required (present Rd when no power) |

Field Code Changed

**Formatted:** Font: 10 pt, Not Bold, Font color: Auto, Do not check spelling or grammar

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### 3.8 Watchdog Timer Requirements (Optional Normative)

Formatted: Heading 2

It is recommended a watchdog timer, which monitors the TCPM-to-TCPC for lack of communication, be implemented by a TCPC if it supports sourcing or sinking voltages greater than vSafe5V. A watchdog timer shall be implemented when PD_INTERFACE_REV register.bcd USB-PD Inter-Block Specification = 0001 0001b and DEVICE_CAPABILITIES_2.WatchdogTimer = 1b.

The watchdog timer functionality shall be enabled whenever TCPC_CONTROL.EnableWatchdogTimer is set to 1b. The watchdog timer shall start when any of the interrupts that are not masked in the Alert register are set or when the Alert# pin is asserted. The watchdog timer is cleared on an I2C access by the TCPM (either read or write). If the ALERT# pin is still asserted after this I2C access, the watchdog timer will reinitialize and start monitoring again until all of the Alerts are cleared or until the ALERT# pin is de-asserted.

Formatted: PARAGRAPH

#### 3.8.1 Watchdog Timer Function

When enabled, the watchdog timer shall start when any unmasked alert bit is set.

An unresponsive TCPM which is unable to clear interrupts within tHVWatchdog, Table 4-45, will cause the watchdog timer to expire. When the watchdog timer expires, the TCPC shall immediately disconnect the CC terminations by setting ROLE_CONTROL bits 3..0 to 1111b, disconnect the Source or Sink paths, discharge VBUS to vSafe0V, and set FAULT_STATUS.I2CInterfaceError.

Any further changes on VBUS need to be initiated by the TCPM when its communication link with the TCPC is restored.

A TCPC shall disable the watchdog timer whenever TCPC_CONTROL.EnableWatchdogTimer is set to 0b.

Formatted: PARAGRAPH

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

## 4 USB Type-C Port Controller Interface

The USB Type-C Port Controller Interface (TCPCI) is a low level interface which handles VBUS and VCONN power connections, CC communication and USB-PD message delivery through a simple register interface. Communication between the TCPC and the USB Type-C Port Manager (TCPM) is over an I2C bus.

The TCPCI uses the I2C protocol with the following behaviors:

1. The TCPM is the only master on the I2C bus.
2. The TCPC is a slave device on the I2C bus.
3. The TCPM designer must meet the I2C bus loading requirements when determining the maximum number of devices on the I2C bus.
4. Each USB Type-C port has its own unique I2C slave address. The TCPC may support multiple USB Type-C ports. In case the TCPC supports multiple ports, each USB Type-C port shall have a unique I2C slave address.
5. The TCPC shall support Fast-mode (Fm+) bus speed. It may also support other bus speeds.
6. The TCPC shall have an open drain output, active low Alert# Pin. This pin is used to indicate a change of state, where Alert# pin is asserted when any Alert Bits are set.
7. The TCPCI shall support an I/O voltage range from 1.8V to 3.6V.
8. The TCPC as a slave device shall be accessible through I2C communication protocols compliant with "I2C-bus specification and user manual Rev.6" (4th April 2014) http://www.nxp.com/documents/user_manual/UM10204.pdf
9. The TCPC should auto-increment the I2C internal register address of the last byte transferred during a read independent of an ACK/NAK from the master.
10. The TCPC may implement the SMBus version 3 bus protocol (Section 6.5 of the SMBus Specification, version 3.0 available at http://smbus.org/specs/).
11. The TCPC shall allow reads to every register even when defined as Write only. The TCPM should assume the register information returned from a Write only register is not valid.
12. The TCPC shall not NAK if the TCPM writes to a register or bit that is not implemented or reserved.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### 4.1 Register Map

The 16-bit registers are used for notation convenience. Each 16-bit register occupies two contiguous bytes, with its 8 Least Significant bits stored in the first (lower address) byte and its 8 Most Significant bits stored in the second (higher address) byte.

**Table 4-~~14~~ ~~1~~. Register Map**

| Address | Register Name | Required /Optional? | Type | Reset Value | Definition |
|---|---|---|---|---|---|
| 00h..01h | VENDOR_ID | Required | R | VD | Table 4-2. VENDOR_ID Register Definition |
| 02h..03h | PRODUCT_ID | Required | R | VD | Table 4-3. PRODUCT_ID Register Definition |
| 04h..05h | DEVICE_ID | Required | R | VD | Table 4-4. DEVICE_ID Register Definition |
| 06h,,07h | USBTYPEC_REV | Required | R | VD | Table 4-5. USBTYPEC_REV Register Definition |
| 08h..09h | USBPD_REV_VER | Required | R | VD | Table 4-6. USBPD_REV_VER Register Description |
| 0Ah..0Bh | PD_INTERFACE_REV | Required | R | VD | Table 4-7. PD_INTERFACE_REV Register Description |
| 0C..0Fh | Reserved | Required | | | Intentionally Blank |
| 10h..11h | ALERT | Required | R/W | 0000h | Table 4-8. ALERT Register Definition |
| 12h..13h | ALERT_MASK | Required | R/W | 0FFFh | Table 4-9. ALERT_MASK Register Definition |
| 14h | POWER_STATUS_MASK | Required | R/W | FFh | Table 4-10. POWER_STATUS_MASK Register Definition |
| 15h | FAULT_STATUS_MASK | Required | R/W | ~~7Fh~~FFh | Table 4-11. FAULT_STATUS_MASK Register Definition |
| 16..17h | Reserved | Required | | | Intentionally Blank |
| 18h | CONFIG_STANDARD_OUTPUT | Optional | R/W | 60h | Table 4-12. CONFIG_STANDARD_OUTPUT Register Definition |
| 19h | TCPC_CONTROL | Required | R/W | 00h | Table 4-13. TCPC_Control Register Definition |
| 1Ah | ROLE_CONTROL | Required | R/W | ~~Table 4-15~~Table 4-15 | Table 4-14. ROLE_CONTROL Register Definition |
| 1Bh | FAULT_CONTROL | Partial R | R/W | 00h | Table 4-16. FAULT_CONTROL Register Definition |
| 1Ch | POWER_CONTROL | Partial R | R/W | ~~1~~60h | Table 4-17. POWER_CONTROL Register Definition |
| 1Dh | CC_STATUS | Required | R | | Table 4-20. CC_STATUS Register Definition |
| 1Eh | POWER_ STATUS | Required | R | | Table 4-21. POWER_STATUS Register Definition |
| 1Fh | FAULT_STATUS | Required | R/W | 80h | Table 4-22. FAULT_STATUS Register Definition |
| 20h..22h | Reserved | Required | R | | Intentionally Blank |
| 23h | COMMAND | Required | W | 00h | Table 4-23. COMMAND Register Definition |

Formatted Table
Formatted: Font: +Body (Cambria), 7 pt
Formatted: Font: +Body (Cambria), 7 pt
Formatted: Font: +Body (Cambria), 7 pt
Formatted: Font: +Body (Cambria), 7 pt
Formatted: Font: +Body (Cambria), 7 pt
Formatted: Font: 7 pt
Formatted: Font: +Body (Cambria), 7 pt
Formatted: TABLE-cell, Centered
Formatted: Font: 7 pt
Formatted: Font: +Body (Cambria), 7 pt
Formatted: Font: +Body (Cambria), 7 pt
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted Table
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted: Centered
Formatted: Font: 7 pt
Formatted: Font: +Body (Cambria), 7 pt

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

| Address | Register Name | Required /Optional? | Type | Reset Value | Definition |
|---|---|---|---|---|---|
| 24h..25h | DEVICE_CAPABILITIES_1 | Required | R | VD | Table 4-24. DEVICE_CAPABILITIES_1 Register Definition~~Table 4-24. DEVICE_CAPABILITIES_1 Register Definition~~ |
| 26h..27h | DEVICE_CAPABILITIES_2 | Required | R | VD | Table 4-25. DEVICE_CAPABILITIES_2 Register Definition~~Table 4-25. DEVICE_CAPABILITIES_2 Register Definition~~ |
| 28h | STANDARD_INPUT_CAPABILITIES | Required | R | VD | Table 4-26. STANDARD_INPUT_CAPABILITIES Register Definition~~Table 4-26. STANDARD_INPUT_CAPABILITIES Register Definition~~ |
| 29h | STANDARD_OUTPUT_CAPABILITIES | Required | R | VD | Table 4-27. STANDARD_OUTPUT_CAPABILITIES Register Definition~~Table 4-27. STANDARD_OUTPUT_CAPABILITIES Register Definition~~ |
| 2Ah..2Dh | Reserved | Required | R | | Intentionally Blank |
| 2Eh | MESSAGE_HEADER_INFO | Required | R/W | Table 4-15~~Table 4-15~~ | Table 4-28. MESSAGE_HEADER_INFO Register Definition~~Table 4-28. MESSAGE_HEADER_INFO Register Definition~~ |
| 2Fh | RECEIVE_DETECT | Required | R/W | 00h | Table 4-29. RECEIVE_DETECT Register Definition~~Table 4-29. RECEIVE_DETECT Register Definition~~ |
| RECEIVE_BUFFER ~~RECEIVE_BUFFER~~ | | | | | |
| 30h | RECEIVE_BYTE_COUNT | Required | R | 00h | Number of Bytes in the RECEIVE_BUFFER that are not stale Table 4-30 |
| 31h | RX_BUF_FRAME_TYPE | Required | R | 00h | Type of received frame (with a reference to a description of the register) |
| 32h | RX_BUF_HEADER_BYTE_0 | Required | R | 00h | Byte 0 (bits 7..0) of message header |
| 33h | RX_BUF_HEADER_BYTE_1 | Required | R | 00h | Byte 1 (bits 15..8) of message header |
| 34h | RX_BUF_OBJ1_BYTE_0 | Required | R | 00h | Byte 0 (bits 7..0) of 1st data object |
| 35h | RX_BUF_OBJ1_BYTE_1 | Required | R | 00h | Byte 1 (bits 15..8) of 1st data object |
| 36h | RX_BUF_OBJ1_BYTE_2 | Required | R | 00h | Byte 2 (bits 23..16) of 1st data object |
| 37h | RX_BUF_OBJ1_BYTE_3 | Required | R | 00h | Byte 3 (bits 31..24) of 1st data object |
| 38h | RX_BUF_OBJ2_BYTE_0 | Required | R | 00h | Byte 0 (bits 7..0) of 2nd data object |
| … | RX_BUF_OBJn_BYTE_m | Required | R | 00h | Byte m of nth data object |
| 4Fh | RX_BUF_OBJ7_BYTE_3 | Required | R | 00h | byte 3 (bits 31..24) of 7th data object |
| | | | | | |
| 50h | TRANSMIT | Required | R/W | 00h | Table 4-34. TRANSMIT Register Definition~~Table 4-34. TRANSMIT Register Definition~~ |
| TRANSMIT_BUFFER | | | | | |
| 51h | TRANSMIT_BYTE_COUNT | Required | R/W | 00h | The number of bytes the TCPM will write |
| 52h | TX_BUF_HEADER_BYTE_0 | Required | R/W | 00h | Byte 0 (bits 7..0) of message header |
| 53h | TX_BUF_HEADER_BYTE_1 | Required | R/W | 00h | Byte 1 (bits 15..8) of message header |
| 54h | TX_BUF_OBJ1_BYTE_0 | Required | R/W | 00h | Byte 0 (bits 7..0) of 1st data object |
| … | TX_BUF_OBJn_BYTE_m | Required | R/W | 00h | Byte m of nth data object |
| 6Fh | TX_BUF_OBJ7_BYTE_3 | Required | R/W | 00h | byte 3 (bits 31..24) of 7th data object |
| | | | | | |
| 70..71h | VBUS_VOLTAGE | Required if greater than vSafe5V | R | 00h | Table 4-38. VBUS_VOLTAGE Register Definition~~Table 4-38. VBUS_VOLTAGE Register Definition~~ |

Formatted Table
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted: Font: 7 pt
Formatted: TABLE-cell, Centered
Formatted: Font: Cambria, 7 pt, Not Bold
Formatted Table
Formatted: Font: 7 pt
Formatted Table
Formatted: Font: 7 pt

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

| Address | Register Name | Required /Optional? | Type | Reset Value | Definition |
|---|---|---|---|---|---|
| 72..73h | VBUS_SINK_DISCONNECT_THRESHOLD | Required Sink >vSafe5V | R/W | vSafe5V | Table 4-39. VBUS_SINK_DISCONNECT_THRESHOLD Register Description~~Table 4-39. VBUS_SINK_DISCONNECT_THRESHOLD Register Description~~ |
| 74..75h | VBUS_STOP_DISCHARGE_THRESHOLD | Required Sink >vSafe5V | R/W | vSafe0V | Table 4-40. VBUS_STOP_DISCHARGE_THRESHOLD Register Description~~Table 4-40. VBUS_STOP_DISCHARGE_THRESHOLD Register Description~~ |
| 76..77h | VBUS_VOLTAGE_ALARM_HI_CFG | Required > vSafe5V | R/W | 00h | Table 4-41. VBUS_VOLTAGE_ALARM_HI_CFG Register Description~~Table 4-41. VBUS_VOLTAGE_ALARM_HI_CFG Register Description~~ |
| 78..79h | VBUS_VOLTAGE_ALARM_LO_CFG | Required > vSafe5V | R/W | 00h | Table 4-42. VBUS_VOLTAGE_ALARM_LO_CFG Register Description~~Table 4-42. VBUS_VOLTAGE_ALARM_LO_CFG Register Description~~ |
| 7A..7Fh | Reserved | Required | | | Intentionally Blank |
| 80h..FF | Vendor defined bits | Optional | | VD | As many as Vendor Defines |

Notes:

VD - Vendor Defined

Formatted Table

Formatted: Font: 7 pt

Formatted: Font: 7 pt

Formatted: Font: 7 pt

Formatted: Font: 7 pt

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### 4.2 TCPC SMBus Optional Normative Requirements

Each row of the register map table defines a register. Some TCPC-s implement the SMBus protocol on top of the I2C protocol and some do not. A TCPC that implements the SMBus protocol requires the TCPM to:

- Read/Write only a single register in a given I2C transaction.
- Write the complete register in a single I2C transaction.
- Begin reading a register from its first byte.

A TCPC that does not implement the SMBus protocol does not enforce these requirements.

If the TCPM attempts to write multiple registers in a single I2C transaction, the TCPC may assert ALERT.InterfaceError and ignore the write. If the TCPM attempts to read multiple registers in a single transaction, the TCPC may assert ALERT.InterfaceError and leave the SDA line open so the TCPM reads all 1's in the following bytes. If the TCPM attempts to read a register address that is not the first byte in that register, the TCPC may assert ALERT.InterfaceError and leave the SDA line open so the TCPM reads all 1's.

### 4.3 Writing and Reading TCPC Registers

#### 4.3.1 Writing Single Byte Registers

To ensure correct behavior with TCPCs that implement SMBus, the TCPC cannot use the I2C short-cut to write consecutive registers in a single operation. The TCPM shall use two transactions to write ROLE_CONTROL and POWER_CONTROL as shown in Figure 4-1.

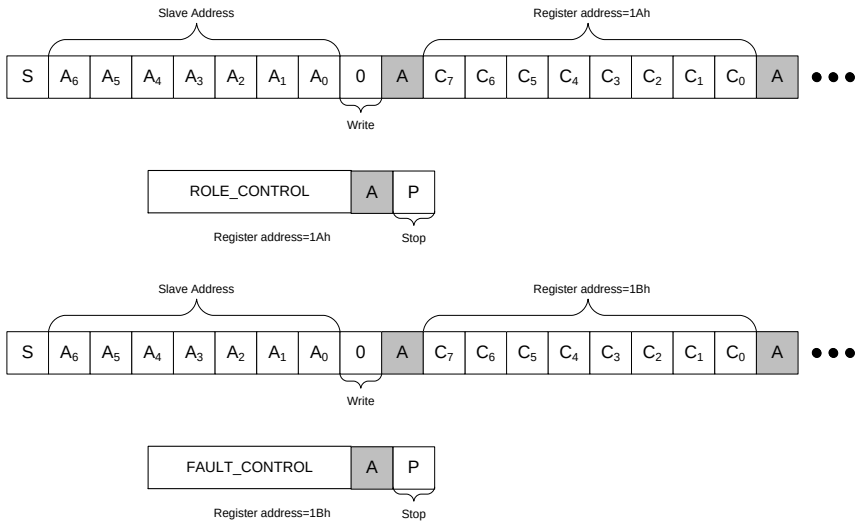



**Figure 4-~~14-1~~. Writing Consecutive Registers with or without the SMBUS Protocol**

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

### 4.3.2 Reading Single Byte Registers

Figure 4-2 indicates how to read single byte registers with I2C or SMBus protocols.

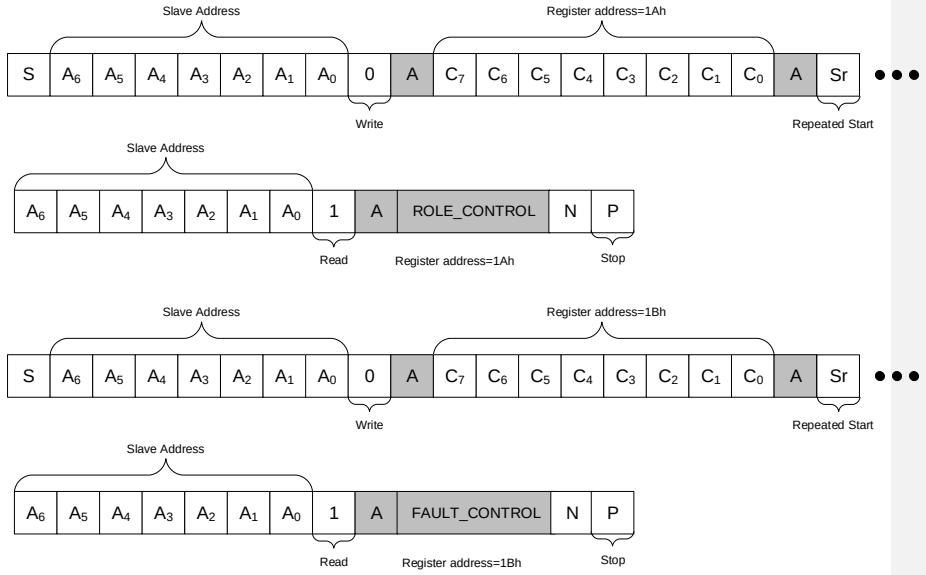



**Figure 4-~~2~~4-~~2~~. Reading Consecutive Registers with or without the SMBus Protocol**

### 4.3.3 Writing Multiple-Byte Registers

Bytes that are always written at the same time can be grouped into a register so the SMBus protocol has less overhead. The TCPM shall write both bytes in the ALERT register at the same time as depicted in the following Figure 4-3.

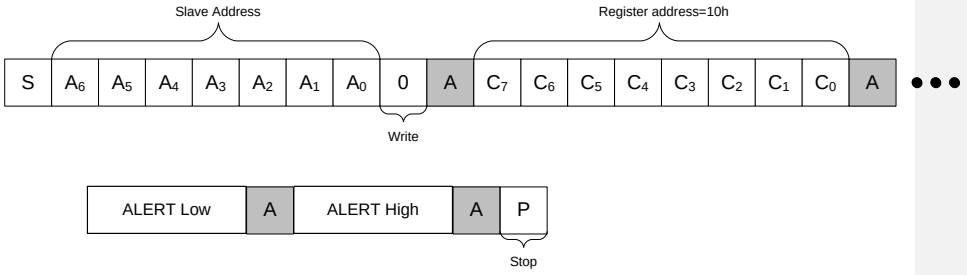



**Figure 4-~~3~~4-~~3~~. Writing a 2-Byte Register with or without the SMBus Protocol**

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

### 4.3.4 Reading Multiple-Byte Registers

Bytes that are always read at the same time can be grouped into a register so the SMBus protocol has less overhead. The TCPM shall read both bytes in the VENDOR_ID register at the same time as depicted in the following Figure 4-4.

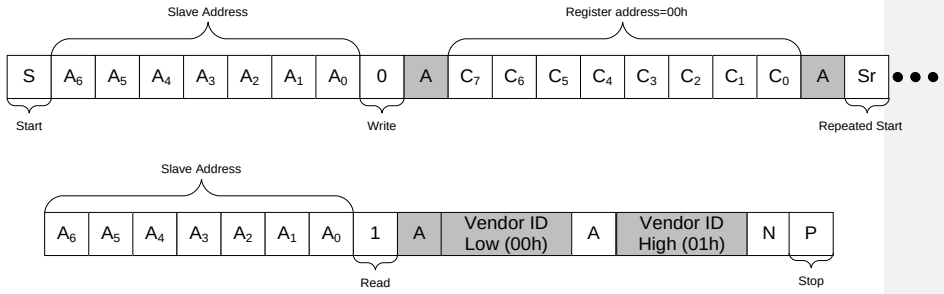



**Figure 4-~~4~~4-~~4~~4. Reading a 2-Byte Register with or without the SMBus Protocol**

### 4.3.5 Writing the TRANSMIT_BUFFER

Figure 4-5 illustrates how the transmit buffer can be written with or without the SMBus protocol.

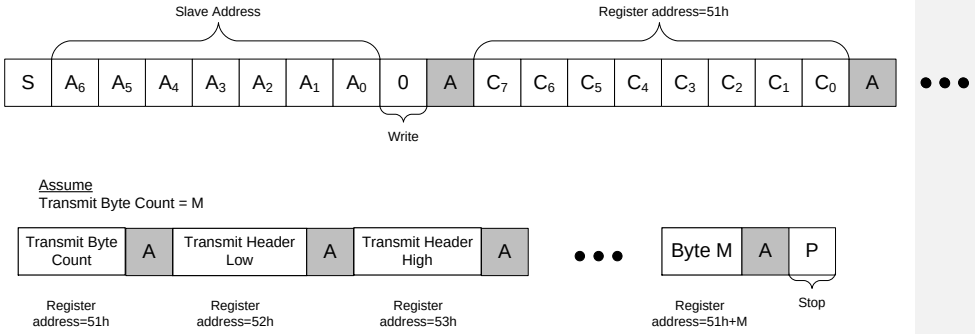



**Figure 4-~~5~~4-~~5~~5. Writing the TRANSMIT_BUFFER with or without the SMBus Protocol**

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

### 4.3.6 Reading the RECEIVE_BUFFER

Figure 4-6 illustrates how the receive buffer can be read with or without the SMBus protocol.

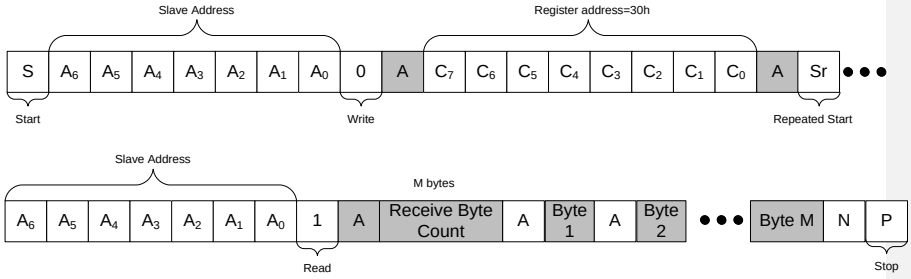



**Figure 4-~~64-6~~. Reading the RECEIVE_BUFFER with or without the SMBus Protocol**

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### 4.4 Register Definition

This section defines the registers for the TCPC.

#### 4.4.1 Identification Registers

##### 4.4.1.1 VENDOR_ID (Required)

A Vendor ID, or VID, is used to identify the TCPC vendor. The VID is a unique 16-bit unsigned integer assigned by USB-IF.

**Table 4-~~2~~4-~~2~~. VENDOR_ID Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B15..0 | Vendor ID (VID) | A unique 16-bit unsigned integer assigned by the USB-IF to the Vendor. |

##### 4.4.1.2 PRODUCT_ID and DEVICE_ID (Required)

The Product ID, or PID, is used to identify the product. The Device ID, bcdDevice, is used to identify the release version of the product. Manufacturers should set the USB Product ID field to a unique value across all USB products from the vendor. The Product ID should identify the product from the vendor and the bcdDevice field should reflect a version number relevant to the release version of the product.

**Table 4-~~3~~4-~~3~~. PRODUCT_ID Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B15..0 | USB Product ID (PID) | A unique 16-bit unsigned integer assigned uniquely by the Vendor to identify the TCPC. |

**Table 4-~~4~~4-~~4~~. DEVICE_ID Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B15..0 | bcdDevice | A unique 16-bit unsigned integer assigned by the Vendor to identify the version of the TCPC. |

##### 4.4.1.3 USBTYPEC_REV (Required)

This register refers to ~~*USB Type-CUSB Type-C*~~ Cable and Connector Specification Revision, ~~*USB Type-CUSB Type-C*~~ represented by a unique 16-bit unsigned register. The format is packed binary coded decimal.

This specification revision 1.0 aligns with USB Type-C Version 1.~~1~~2.

**Table 4-~~5~~4-~~5~~. USBTYPEC_REV Register Definition (Required)**

| Bit(s) | Name | Description |
|---|---|---|
| B15..8 | Reserved | Set to 0 |
| B7..0 | ~~*USB Type-CUSB Type-C*~~ Revision | Revision number assigned by USB-IF |

**Formatted:** bookmark Char, Font: Not Bold

**Formatted:** bookmark Char, Font: Not Bold

**Formatted:** bookmark Char, Font: 8 pt, Not Bold

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.1.4 USBPD_REV_VER (Required)

This register refers to USB-PD Specification Revision and Version, ***USB PDUSB PD*** represented by a unique 16-bit unsigned integer. The format is packed binary coded decimal.

Formatted: bookmark Char, Font: Not Bold

This specification revision 1.0 aligns with USB-PD Revision 2.0 version 1.12.

**Table 4-64-6. USBPD_REV_VER Register Description**

| Bit(s) | Name | Description |
|---|---|---|
| B15..8 | bcdUSBPD Revision | 0010 0000 – Revision 2.0 |
| B7..0 | bcdUSBPD Version | 0001 0000 – Version 1.0<br>0001 0001 0010 – Version 1.12<br>Etc. |

#### 4.4.1.5 USB-PD Port Controller Interface Inter-Block Specification Revision (Required)

The USB-PD Port Controller Inter-Block Specification Revision register refers to this Specification Revision and Version represented by a unique 16-bit unsigned integer. The format is packed binary coded decimal.

**Table 4-74-7. PD_INTERFACE_REV Register Description**

| Bit(s) | Name | Description |
|---|---|---|
| B15..8 | bcd USB-PD Inter-Block Specification Revision | 0001 0000 0000 – Revision 1.0 0 (this release) |
| B7..0 | bcd USB-PD Inter-Block Specification Version | 0001 0000 – Version 1.0 (previous release)<br>0001 0000 0001 – Version 1.0 1 (this release)<br>0001 0001 – Version 1.1<br>Etc. |

Copyright © 2015 2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.2 ALERT Register (Required)

This register is set by TCPC and cleared by TCPM.

This register is used to communicate a status change from the TCPC to the TCPM. After an event or condition occurs, the TCPC shall set the corresponding bit in the ALERT register. The TCPC shall keep the bit associated with the ALERT asserted until the TCPM writes a 1 to clear it.

The TCPC indicates an alert status change has occurred by presenting a logical 1 in the corresponding alert bit position in this register and asserting the Alert# pin. The TCPM clears the ALERT bit by writing a logical 1 to the respective ALERT bit position. The TCPM can clear any number of ALERT bits in a single write by setting multiple bits to logical 1 and the rest of the bits in the register to logical 0. The TCPM writing a logical 0 to any ALERT bit has no effect, and therefore does not cause those ALERT bits to be set or cleared. The Alert# pin remains asserted until all ALERT bits are cleared by the TCPM. If the TCPM writes a logical 1 to a bit that is already logical 0, the TCPC shall not change the value of that bit.

Writing a 1 to ALERT.RxBufferOverflow does not clear it unless the TCPM also writes a 1 to ALERT.ReceiveSOP*MessageStatus. The ALERT.RxBufferOverflow is always asserted if the SOP* buffer registers are full, and those registers can only be cleared by writing a 1 to ALERT.ReceiveSOP*MessageStatus.

**Table 4-~~8~~4-~~8~~. ALERT Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B15 | Vendor Defined Alert | 0b: Cleared<br>1b: A vendor defined alert has been detected. Defined in the VENDOR_DEFINED registers. Refer to the vendor datasheet for details.<br>This bit can be cleared, regardless of the current status of the alert source. |
| B1~~4~~5..12 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B11 | VBUS Sink Disconnect Detected | 0b: Cleared<br>1b: A VBUS Sink Disconnect Threshold crossing has been detected |
| B10 | Rx Buffer Overflow | 0b: TCPC Rx buffer is functioning properly<br>1b: TCPC Rx buffer has overflowed. Future GoodCRC shall not be sent.<br>Writing 1 to this register acknowledges the overflow. The overflow is cleared by writing to ALERT.ReceiveSOP*MessageStatus |
| B9 | Fault | 0b: No Fault<br>1b: A Fault has occurred. Read the FAULT_STATUS register |
| B8 | VBUS Voltage Alarm Lo | 0b: Cleared<br>1b: A low-voltage alarm has occurred |
| B7 | VBUS Voltage Alarm Hi | 0b: Cleared<br>1b: A high-voltage alarm has occurred |
| B6 | Transmit SOP* Message Successful | 0b: Cleared,<br>1b: Reset or SOP* message transmission successful. GoodCRC response received on SOP* message transmission. Transmit SOP* message buffer registers are empty. |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

| Bit(s) | Name | Description |
|---|---|---|
| B5 | Transmit SOP* Message Discarded | 0b: Cleared,<br>1b: Reset or SOP* message transmission not sent due to incoming receive message. Transmit SOP* message buffer registers are empty. |
| B4 | Transmit SOP* Message Failed | 0b: Cleared,<br>1b: SOP* message transmission not successful, no GoodCRC response received on SOP* message transmission. Transmit SOP* message buffer registers are empty. |
| B3 | Received Hard Reset | 0b: Cleared,<br>1b: Received Hard Reset message |
| B2 | Receive SOP* Message Status | 0b: Cleared,<br>1b: RECEIVE_BUFFER register changed. RECEIVE_BYTE_COUNT being set to 0 does not set this bit. |
| B1 | Power Status | 0b: Cleared, 1b: Power Status changed |
| B0 | CC Status | 0b: Cleared, 1b: CC Status changed |

Note: The TCPM is not expected to mask the “Received Hard Reset” alert bit.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### 4.4.3 Mask Registers

The registers in this section define the masks that may be set for the ALERT registers. A masked register will still indicate in the ALERT register, but shall not set the Alert# pin low. POWER_STATUS_MASK and FAULT_STATUS_MASK registers are nested Alerts. A POWER_STATUS change has to be unmasked in both the POWER_STATUS_MASK and the ALERT.PowerStatusInterruptMask to enable the Alert# pin. A FAULT_STATUS change has to be unmasked in both the FAULT_STATUS_MASK and the ALERT.PowerStatusInterruptMask to enable the Alert# pin.

#### 4.4.3.1 ALERT_MASK (Required)

This is an event interrupt mask. It is masked and unmasked by the TCPM. The ALERT_MASK Register is cleared by the TCPM. The ALERT_MASK Register shall be initialized to 0FFFh upon power on or Hard Reset.

The assertion of Alert# pin is prevented when the corresponding bit in this register is set to zero by the TCPM. Setting any bits in this register has no effect on ALERT registers.

**Table 4-~~9~~4-~~9~~. ALERT_MASK Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B15 | Vendor Defined Alert | 0b: Interrupt masked 1b: Interrupt unmasked |
| B1~~5~~4..12 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B11 | VBUS Sink Disconnect Detected | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B10 | Rx Buffer Overflow | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B9 | Fault | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B8 | VBUS Voltage Alarm Lo | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B7 | VBUS Voltage Alarm Hi | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B6 | Transmit SOP* Message successful Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B5 | Transmit SOP* Message discarded Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B4 | Transmit SOP* Message failed Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B3 | Received Hard Reset Message Status Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked<br>(The Hard Reset should generally not be masked) |
| B2 | Receive SOP* Message Status Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B1 | Power Status Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B0 | CC Status Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.3.2 POWER_STATUS_MASK (Required)

This is an event interrupt mask. It is masked and unmasked by the TCPM. This register allows individual masking of power events. The POWER_STATUS_MASK Register is cleared by the TCPM. The POWER_STATUS_MASK Register shall be initialized per Table 4-1 upon power on or Hard Reset.

The assertion of Alert# pin is prevented when the corresponding bit is set to zero by the TCPM. Setting any bits in this register has no effect on the ALERT or POWER_STATUS registers. ALERT.PowerStatus is asserted if the bit-wise AND of POWER_STATUS and POWER_STATUS_MASK results in any bits that have the value 1.

**Table 4-104-10. POWER_STATUS_MASK Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7 | Debug Accessory Connected | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B6 | TCPC Initialization Status Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B5 | Sourcing High Voltage Status Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B4 | Sourcing VBUS Status Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B3 | VBUS Present Detection Status Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B2 | VBUS Present Status Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B1 | VCONN Present Status Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B0 | Sinking VBUS Status Interrupt Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |

#### 4.4.3.3 FAULT_STATUS_MASK (Required)

This is an event interrupt mask. It is masked and unmasked by the TCPM. This register allows individual masking of fault events. The FAULT_STATUS_MASK Register is cleared by the TCPM. The FAULT_STATUS_MASK Register shall be initialized per Table 4-1 upon power on or Hard Reset.

The assertion of Alert# pin is prevented when the corresponding bit is set to zero by the TCPM. ALERT.FaultStatus is asserted if the bit-wise AND of FAULT_STATUS and FAULT_STATUS_MASK results in any bits that have the value 1.Setting any bits in this register has no effect on the ALERT or FAULT_STATUS registers.

Over current protection, OCP, can be either integrated or external to the TCPC. An external OCP fault signal may be connected to the STANDARD_INPUT_SIGNALS Vbus external over current fault and the status reported in this register. An internal OCP fault shall be reported in this register if implemented. OCP is defined in a vendor defined manner.

Over voltage protection, OVP, can be either integrated or external to the TCPC. An external OVP fault signal may be connected to the STANDARD_INPUT_SIGNALS Vbus external over voltage fault and the status reported in this register. An internal OVP fault shall be reported in this register if implemented. OVP is defined in a vendor defined manner.

Copyright © 2015-2016 USB 3.0 Promoter Group. All rights reserved.

**Table 4-114-11. FAULT_STATUS_MASK Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7 | AllRegistersResetToDefaultReserved | 0b: Interrupt masked, 1b: Interrupt unmaskedShall be set to zero by sender and ignored by receiver |
| B6 | Force Off Vbus Interrupt Status Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B5 | Auto Discharge Failed Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B4 | Force Discharge Failed Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B3 | Internal or External OCP<br>Vbus Over Current Protection Fault Interrupt Status Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B2 | Internal or External OVP<br>Vbus Over Voltage Protection Fault Interrupt Status Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B1 | Vconn Over Current Fault Interrupt Status Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |
| B0 | I2C Interface Error Interrupt Status Mask | 0b: Interrupt masked, 1b: Interrupt unmasked |

Copyright © 2015-2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.4 CONFIGURE STANDARD OUTPUT (Optional Normative)

This register is required if it is reported as supported in the one of the DEVICE_CAPABILITES registers (Section 4.4.8.1). This read/write register is used to configure the Standard Outputs or read the status of the Standard Outputs. The TCPM writes to this register to set the Standard Output Signal defined in Table 4-44. The Standard Outputs shall reset to open-drain per Table 4-1.

**Table 4-~~12~~4-~~12~~. CONFIG_STANDARD_OUTPUT Register Definition**

| Bit(s) | Name | Type |
|---|---|---|
| B7 | High Impedance outputs | 0b: Standard output control (default)<br>1b: Force all outputs to high impedance<br>May be used to save power in Sleep<br>Controlled by the TCPM. |
| B6 | Debug Accessory Connected# | 0b: Debug Accessory Connected# output is driven low. Debug Accessory connected<br>1b: Debug Accessory Connected# output is driven high. No Debug Accessory connected (default)<br>Controlled by either the TCPM or TCPC. The TCPC shall ignore writes to this bit if TCPC_CONTROL.DebugAccessoryControl = 0b. |
| B5 | Audio Accessory Connected# | 0b: Audio Accessory connected<br>1b: No Audio Accessory connected (default)<br>Controlled by the TCPM |
| B4 | Active Cable Connected | 0b: No Active Cable connected (default)<br>1b: Active Cable connected<br>Controlled by the TCPM |
| B3..2 | MUX Control | 00b: No connection (default)<br>01b: USB3.1 Connected<br>10b: DP Alternate Mode – 4 lanes<br>11b: USB3.1 + Display Port Lanes 0 & 1<br>Controlled by the TCPM |
| B1 | Connection Present | 0b: No Connection (default)<br>1b: Connection<br>Controlled by the TCPM. |
| B0 | Connector Orientation | 0b: Normal (CC1=A5, CC2=B5, TX1=A2/A3, RX1=B10/B11) default<br>1b: Flipped (CC2=A5, CC1=B5, TX1=B2/B3, RX1=A10/A11)<br>Controlled by the TCPM<br>The TCPC shall ignore writes to this bit if TCPC_CONTROL.DebugAccessoryControl = 0 |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### 4.4.5 Control and Configuration Registers

#### 4.4.5.1 TCPC_CONTROL (Required)

The TCPM writes to the TCPC_ CONTROL register to set the Plug Orientation and enable/disable clock stretching.

I2C_Clock_Stretching_Control allows the TCPM to control TCPC clock stretching on the I2C bus. Allowing clock stretching may result in lower power from the TCPC, but degrade throughput. Disabling clock stretching will result in increased I2C bus throughput, but may result in higher TCPC power. The TCPC is not allowed to NAK I2C transfers no matter which clock stretching setting is chosen by the TCPM, unless the TCPM has put it to sleep using COMMAND.I2CIdle.

**Table 4-~~13~~4-~~13~~. TCPC_Control Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7..~~5~~6 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B5 | Enable Watchdog Timer | 0b: Watchdog Monitoring is disabled (default)<br>1b: Watchdog Monitoring is enabled<br>Enables Watchdog Timer Monitoring according to Section 3.8<br>Required if DEVICE_CAPABILITIES_2.Watch Dog Timer = 1b |
| B4 | Debug Accessory Control | 0b: Controlled by TCPC (power on default)<br>1b: Controlled by TCPM. The TCPM writes 1b to this register to take over control of asserting the DebugAccessoryConnected#. See ~~Figure 4-13~~.<br>Required (Register is required but output is not required) |
| B3..2 | I2C Clock Stretching Control | Clock Stretching Control<br>00b: Disable clock stretching. TCPC shall not perform any clock stretching during I2C transfers.<br>01b: Reserved<br>10b: Enable clock stretching. TCPC is allowed limited clock stretching during each I2C Transfer.<br>11b: Enable clock stretching only if the Alert pin is not asserted. As soon as Alert is asserted, clock stretching is disabled by the TCPC.<br>The TCPC datasheet should contain details as to the power consequences of clock stretching as well as the max duration of clock stretching per I2C transaction. This is only necessary if clock stretching is implemented. The TCPC shall limit total clock stretching as detailed in Section 4.10.<br>This feature is optional. The TCPC is allowed to ignore updates to this bit field if it has not implemented clock stretching. The power on default value is such to disable clock stretching. |

Field Code Changed

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

| Bit(s) | Name | Description |
|---|---|---|
| B1 | BIST Test Mode | Setting this bit to 1 is intended to be used only when a USB compliance tester is using USB BIST Test Data to test the PHY layer of the TCPC. The TCPM should clear this bit when a detach is detected.<br><br>0: Normal Operation. Incoming messages enabled by RECEIVE_DETECT passed to TCPM via Alert.<br>1: BIST Test Mode. Incoming messages enabled by RECEIVE_DETECT result in GoodCRC response but may not be passed to the TCPM via Alert. TCPC may temporarily store incoming messages in the Receive Message Buffer, but this may or may not result in a Receive SOP* Message Status or a Rx Buffer Overflow alert.<br><br>The TCPM can mask or ignore received message alerts when this bit is set to 1 since the TCPC may or may not assert the alert. The TCPM may also treat received messages in this mode in the same way as received messages during normal operation. |
| B0 | Plug Orientation | 0b: When Vconn is enabled, apply it to the CC2 pin. Monitor the CC1 pin for BMC communications if PD messaging is enabled.<br>1b: When Vconn is enabled, apply it to the CC1 pin. Monitor the CC2 pin for BMC communications if PD messaging is enabled.<br>Required |

#### 4.4.5.2 ROLE_CONTROL (Required)

The TCPM writes to this register to configure the CC pull up (Rp) or pull down (Rd) resistors.

Table 4-15 defines the power on default for ROLE_CONTROL and MESSAGE_HEADER_INFO.

The TCPM shall ~~first~~ write B6 (DRP) = 0b and B3..0 (CC1/CC2) if it wishes to control the Rp/Rd directly instead of having the TCPC perform DRP toggling autonomously. ~~In this mode~~ When controlling Rp/Rd directly, the TCPM writes to B3..0 (CC1/CC2) each time it wishes to change the CC1/CC2 values.

~~The TCPM may configure the TCPC to autonomously perform a static search for connection (no toggle) by first writing B6 (DRP) = 0b and the static value of Rp/Rd to B3..0 (CC1/CC2). An example of this~~ This would be commonly used for TCPM-TCPC implementing ~~is for~~ Source or Sink only as well as ~~conditions. Another example is~~ when a connection has been detected via DRP toggling but the TCPM wishes to attempt Try.Src or Try.Snk scenario.

The TCPM may configure the TCPC to autonomously toggle the Rp/Rd when the TCPM-TCPC is implementing a DRP. When initiating autonomous DRP toggling, ~~. T~~the TCPM shall write B6 (DRP) =1b and the starting value of Rp/Rd to B3..0 (CC1/CC2) to indicate DRP autonomous toggling mode to the TCPC. The TCPC shall not start the DRP toggling until subsequently the TCPM writes to the COMMAND register to start the DRP toggling or there is a change in POWER_CONTROL.AutoDischargeDisconnect as in ~~. Refer to~~ Figure 4-11~~Figure 4-11~~. It is recommended the TCPM write ROLE_CONTROL.DRP=0 before writing to POWER_CONTROL.AutoDischargeDisconnect and start the DRP toggling using COMMAND.Look4Connection as shown in Figure 4-16, Figure 4-17, and Figure 4-18.

If DRP=1b, the only allowed values for CC1/CC2 are Rp/Rp or Rd/Rd. COMMAND.Look4Connection shall do nothing if CC1/CC2 are not Rp/Rp or Rd/Rd.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

When CC1 and CC2 are set to Open and DRP = 0b, the TCPC may power down the PHY and CC Status comparators.

**Table 4-144-14. ROLE_CONTROL Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B6 | DRP | 0b: No DRP. Bits B3..0 determine Rp/Rd/Ra or open settings<br>1b: DRP<br>The TCPC shall use the Rp value defined in B5..4 when a connection is resolved, ie. Upon entry to Potential_Connect_as_Src in Figure 4-11. TCPC State Diagram before a ConnectionFigure 4-11. TCPC State Machine before a Connection.<br>The TCPC toggles CC1 & CC2 after receiving COMMAND.Look4Connection and until a connection is detected. Upon connection, the TCPC shall resolve to either an Rp or Rd and report the CC1/CC2 State in the CC_STATUS register.<br>The CC pins shall stay in Potential_Connect_as_Src Potential_start_as_SRC or Potential_Connect_as_Sink Potential_start_as_Sink until directed otherwise. |
| B5..4 | Rp Value | 00b: Rp default<br>01b: Rp 1.5A<br>10b: Rp 3.0A<br>11b: Reserved |
| B3..2 | CC2 | 00b: Ra<br>01b: Rp (Use Rp definition in B5..4)<br>10b: Rd<br>11b: Open (Disconnect or don't care) |
| B1..0 | CC1 | 00b: Ra<br>01b: Rp (Use Rp definition in B5..4)<br>10b: Rd<br>11b: Open (Disconnect or don't care) |

Table 4-15 defines the power on default for ROLE_CONTROL and MESSAGE_HEADER_INFO.

**Table 4-154-15. Power on Default Conditions**

| DEVICE_CAPABILITIES. RolesSupported | ROLE_CONTROL (Default) | MESSAGE_HEADER_INFO (Default) |
|---|---|---|
| Source or Sink (000b) | 0Ah | 02h |
| Source only (001b) | 05h – Not dead battery<br>N/A – Dead battery | 0Bh |
| Sink only (010b) | 0Ah | 02h |
| Sink with Accessory (011b) | 0Ah | 02h |
| DRP (100b) | 0Ah – dDead battery<br>4Ah – Not dead battery and DebugAccessoryIndicator supported<br>0Fh – Not dead battery and DebugAccessoryIndicator not supported1Fh – non dead battery | 02h |

Formatted Table

Copyright © 2015-2016 USB 3.0 Promoter Group. All rights reserved.

| DEVICE_CAPABILITIES. RolesSupported | ROLE_CONTROL (Default) | MESSAGE_HEADER_INFO (Default) |
|---|---|---|
| Source, Sink, DRP (101b and 110b)<br>Applies to SOP Devices | 0Ah – Source, Sink, or DRP dead battery<br>4Ah – Not dead battery and DebugAccessoryIndicator supported<br>0Fh – Not dead battery and DebugAccessoryIndicator not supported~~1Fh – DRP non dead battery~~ | 02h |

Formatted Table

#### 4.4.5.3 FAULT_CONTROL (Required)

The TCPM writes to FAULT_CONTROL to enable/disable FAULT circuitry.

**Table 4-~~16~~4~~-16~~. FAULT_CONTROL Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7..5 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B4 | Force Off VBUS<br>(Source or Sink) | 0b: Allow STANDARD INPUT SIGNAL Force Off Vbus control (default)<br>1b: Block STANDARD INPUT SIGNAL Force Off Vbus control<br>This enables or disables the STANDARD INPUT SIGNAL Force Off Vbus (4.5.1) functionality for debug purposes. Required if STANDARD_INPUT_CAPABILITES.ForceOffVBUS |
| B3 | VBUS Discharge Fault Detection Timer | 0b: VBUS Discharge Fault Detection Timer enabled<br>1b: VBUS Discharge Fault Detection Timer disabled<br>This enables the timers for both FAULT_STATUTS.AutoDischargeFailed and FAULT_STATUS.ForceDischargeFailed<br>Required |
| B2 | Internal or External OCP<br>VBUS Over Current Protection Fault | 0b: Internal and External OCP circuit enabled<br>1b: Internal and External OCP circuit disabled<br>Required if DEVICE_CAPABILIITES_1.VBUSOCPReporting = 1b |
| B1 | Internal or External OVP<br>VBUS Over Voltage Protection Fault | 0b: Internal and External OVP circuit enabled<br>1b: Internal and External OVP circuit disabled<br>Required if DEVICE_CAPABILIITES_1.VBUSOVPReporting = 1b |
| B0 | VCONN Over Current Fault | 0b: Fault detection circuit enabled<br>1b: Fault detection circuit disabled<br>Required if DEVICE_CAPABILITIES_2.VCONNOvercurrentFaultCapable = 1b |

#### 4.4.5.4 POWER_CONTROL (Required)

The timing parameters for the TCPM in conjunction with the TCPC must meet the ***USB PD~~USB PD~~*** requirements.

Formatted: bookmark Char, Font: Not Bold

The TCPM reads the CC_STATUS, POWER_STATUS and VBUS_VOLTAGE registers to determine the connection state and the orientation of a USB Type-C port.

To source VCONN over one of the CC pins (irrespective of the status of VBUS), all of the following conditions are required:

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

- The TCPM shall set EnableVCONN to logical 1
- The TCPM shall write to TCPC_CONTROL.PlugOrientation to inform TCPC which CC pin (not connected through the cable) is repurposed as VCONN

**Table 4-~~17~~4-~~17~~. POWER_CONTROL Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B6 | VBUS_VOLTAGE Monitor | 0b: VBUS_VOLTAGE Monitoring is enabled ~~(default)~~<br>1b: VBUS_VOLTAGE Monitoring is disabled (default)<br>Controls only VBUS_VOLTAGE Monitoring. VBUS_VOLTAGE shall report all zeroes if disabled.<br>Required if DEVICE_CAPABILITIES_1.VBUSMeasuremen andAlarmCapable = 1b |
| B5 | Disable Voltage Alarms | 0b: Voltage Alarms Power status reporting is enabled ~~(default)~~<br>1b: Voltage Alarms Power status reporting is disabled (default)<br>Controls VBUS_VOLTAGE_ALARM_HI_CFG and VBUS_VOLTAGE_ALARM_LO_CFG.<br>Required if DEVICE_CAPABILITIES_1.VBUSMeasuremen andAlarmCapable = 1b |
| B4 | AutoDischargeDisconnect | 0b: The TCPC shall not automatically discharge VBUS based on VBUS voltage (default)~~.~~<br>1b: The TCPC shall automatically discharge ~~(default)~~<br>Refer to 4.4.5.4.1 and 4.4.5.4.2. Strength of discharge set by tAutoDischarge in Table 4-18<br>Setting this bit in a Source TCPC triggers the following actions upon disconnection detection:<br>1. Disable sourcing power over VBUS<br>2. VBUS discharge<br>Sourcing power over VBUS shall be disabled before or at same time as starting VBUS discharge.<br>Setting this bit in a Sink TCPC triggers the following action upon disconnection detection:<br>1. VBUS discharge<br>The TCPC shall automatically disable discharge once the voltage on VBUS is below vSafe0V (max) or VBUS_STOP_DISCHARGE_THRESHOLD. Disconnect detection is defined in Figure 4-13~~Figure 4-12~~. VBUS_STOP_DISCHARGE_THRESHOLD, if enabled, takes priority over vSafe0V.<br>Required |
| B3 | Enable Bleed Discharge | 0b: Disable bleed discharge (default)<br>1b: Enable bleed discharge of VBUS<br>Bleed Discharge is a low current discharge to provide a minimum load current if needed<br>10K Ohms or 2mA recommended<br>Refer to 4.4.5.4.5<br>Required if DEVICE_CAPABILITIES_1.BleedDischarge = 1b |
| B2 | Force Discharge | 0b: Disable forced discharge (default)<br>1b: Enable forced discharge of VBUS.<br>Refer to 4.4.5.4.3<br>Required if DEVICE_CAPABILITIES_1.ForceDischarge = 1b |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

| Bit(s) | Name | Description |
|---|---|---|
| B1 | VCONN Power Supported | 0b: TCPC delivers at least 1W on VCONN<br>1b: TCPC delivers at least the power indicated in DEVICE_CAPABILITIES.VCONNPowerSupported<br>Refer to TCPC datasheet for actual power limit implemented<br>Required |
| B0 | Enable VCONN | 0b: Disable VCONN Source (default)<br>1b: Enable VCONN Source to CC<br>Required |

**Table 4-~~18~~4-~~18~~. Discharge Timing Parameters**

| Name | Description | Min | Max | Units |
|---|---|---|---|---|
| tAutoDischarge | Time to discharge 100uF and 400 Ohms maximum or 24mA minimum when AutoDischarge is engaged | | 50 | ms |
| tDisconnectDetect | Time from Disconnect to detection of a Disconnect | | 6 | ms |

#### 4.4.5.4.1 Automatic Source Full Discharge by the TCPC after a Disconnect (normative)

When in the Source mode and POWER_CONTROL.AutoDischargeDisconnect=1, the TCPC shall fully discharge VBUS to vSafe5V (max) within tSafe5V and then to vSafe0V within tSafe0V when a Disconnect occurs. The TCPC is in Source mode any time MESSAGE_HEADER_INFO.PowerRole=1. A TCPC in Source mode shall detect a Disconnect if the CCState for the monitored CC pin indicates SRC.Open. The monitored CC pin is specified by TCPC_CONTROL.PlugOrientation.

The TCPC shall discharge Vbus to vSafe0V after a power on reset before applying the Rp.

#### 4.4.5.4.2 Automatic Sink Discharge by the TCPC after a Disconnect (normative)

The TCPC is in the Sink mode any time MESSAGE_HEADER_INFO.PowerRole=0. A TCPC in Sink mode shall use either the POWER_STATUS.VbusPresent transition from 1b to 0b or VBUS falling below VBUS_SINK_DISCONNECT_THRESHOLD, Table 4-39, as a Sink disconnect indicator. The mechanism used shall be defined in DEVICE_CAPABILITIES_2.SinkDisconnect. The Sink TCPC shall detect a cable removal within tDisconnectDetect, Table 4-18, of the Sink disconnect indicator change and enable the automatic discharge circuitry.

Should a TCPM need to know when VBUS discharge is complete, it may set VBUS_VOLTAGE_ALARM_LO_CFG to vSafe0V if DEVICE_CAPABILITIES_2.StopDischargeThreshold=1 or to vSafe0V (max) if DEVICE_CAPABILITIES_2.StopDischargeThreshold=0.

When in Sink mode and POWER_CONTROL.AutoDischargeDisconnect=1, the TCPC shall fully discharge VBUS to vSafe5V (max) within tSafe5V and then to vSafe0V within tSafe0V of the removal of the cable.

When the source is removed, the system load and/or bleed discharge will discharge the sink bulk capacitance cSnkBulkPd. The time required to discharge cSnkBulkPd to below the disconnect detection threshold is tSinkDischargeBleed and depends upon the strength of the bleed discharge and the system load. The time required to discharge the sink bulk capacitance to vSafe5V is tSinkDischargeFull and depends upon the strength of the full discharge and the system load. The total time tSinkDischargeBleed + tSinkDischargeFull

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

shall not exceed tSafe5V (max) to transition to vSafe5V. The total time to reach vSafe0V shall not exceed tSafe0V.

As an example, if:

- the bleed discharge pull down is ~~10kohm~~8.5kohm,
- the sink bulk capacitance cSnkBulkPd is 100uF,
- there is no system load,
- the initial voltage is 21.5 V (20 V + 5% + vSrcValid),
- and VBUS_SINK_DISCONNECT_THRESHOLD = 16V (0.8*20V)

then tSinkDischargeBleed = ~~296ms~~250ms. Once the VBUS_SINK_DISCONNECT_THRESHOLD has been reached, the Source connects its full discharge resistance of 400 Ohms. The Source then discharges to vSafe5V in 46us and vSafe0V in 120us. The total time to reach vSafe0V is then tSinkDischargeBleed + tSinkDischargeFull = 250.05ms. The total time to reach vSafe0V is then 250.12ms. Both tSafe5V and tSafe0V can be met.

~~This means tSinkDischargeFull < 45ms required can be achieved with a full discharge pull down of 400 Ohm.~~

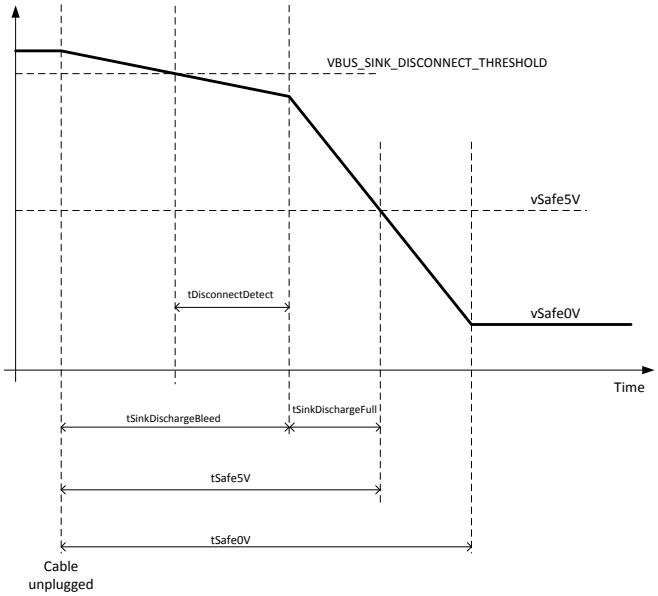



**Figure 4-~~7~~4-~~7~~. Automatic VBUS Sink Discharge by the TCPC after a Disconnect**

#### 4.4.5.4.3 Discharge by the Source TCPC during a Connection (Optional Normative)

While there is a valid Source-to-Sink connection, the TCPC acting as a Source shall discharge VBUS whenever POWER_CONTROL.ForceDischarge=1. The TCPC shall automatically disable discharge once the voltage on VBUS is below the value indicated by VBUS_STOP_DISCHARGE_THRESHOLD. A Source TCPC transitioning from a higher to lower

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

voltage shall disconnect the ForceDischarge in time to meet the ***USB PD~~USB PD~~*** vSrcValid requirement and remove the Force Discharge circuit.

The TCPC shall discharge Vbus to vSafe0V after a power on reset before applying the Rp.

#### 4.4.5.4.4 Discharge by the Sink TCPC during a Connection (Optional Normative)

While there is a valid Source-to-Sink connection, the TCPC acting as a Sink shall reduce its current to less than iSnkSwapStdby within tSnkSwapStby (***USB PD~~USB PD~~***) when handling a Power Role Swap or Hard Reset. The TCPM shall write POWER_CONTROL.AutoDischargeDisconnect to 0 or VBUS_SINK_DISCONNECT_THRESHOLD to 0 and COMMAND.DisableSinkVbus to disable the Sink disconnect detection and remove the Sink connection upon reception of or prior to transmitting a Power Role Swap or Hard Reset.

#### 4.4.5.4.5 Bleed Discharge (Optional Normative)

Bleed discharge is enabled and disabled by the TCPM. The Bleed discharge is a low current discharge for providing a minimal load.

### 4.4.6 Status Registers

These registers indicate the state of the TCPC. These registers are set by the TCPC and read by the TCPM.

The CC_STATUS and POWER_STATUS registers are not latched and are continually updated unless powered off. The FAULT_STATUS register is latched.

#### 4.4.6.1 CC_STATUS (Required)

The TCPC updates this register on a Connect or Disconnect. The TCPC shall update the CC_STATUS register within tSetReg (Table 4-45) of a change in ROLE_CONTROL.DRP or a change on the CC1 or CC2 wires, after debounce.

The TCPM starts the DRP toggling by writing to the COMMAND register.

The TCPM reads this register upon detecting an Alert# and seeing the ALERT.CcStatus=1. The TCPC indicates the Connection status, the Connection result, and the current CC state in this register.

The TCPC shall set CC_STATUS.Looking4Connection = 0b when it has detected a potential connection. The Autonomous DRP toggling details are defined in Section 0.

The TCPM reads the Looking4Connection to determine if the TCPC is toggling Rp/Rd when operating as a DRP. The TCPM reads the CC_STATUS.ConnectResult to determine if a DRP TCPC is presenting an Rp or Rd. The TCPM shall read the CC1State and CC2State to determine the CC1 and CC2 states.

When reporting the state of the CC lines, the TCPC shall debounce for tTCPCfilter. The TCPC shall perform a minimal debounce and the TCPM must complete the debounce as defined in ***USB Type-C~~USB Type-C~~***.

The TCPM as a Source detects a Sink attachment and detachment by reading Cc1State and Cc2State bits. The CC Status monitoring may be disabled per Section 4.8.3.

A TCPM which is using polling rather than Alerts should assume the data in the CC_STATUS register is not valid until at least tCcStatusDelay + tTCPCFilter + tCcTCPCSampleRate (max)

Formatted: bookmark Char, Font: Not Bold

Formatted: bookmark Char, Font: Not Bold

Formatted: bookmark Char, Font: Not Bold

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

(Table 4-45) after the ROLE_CONTROL has been updated. The tCcTCPCSampleRate is the CC sample rate used by the TCPC. The CC sampling method and rate is performed in a vendor specific manner and therefore outside the scope of this specification.

**Table 4-~~19~~4-~~19~~. Debounce requirements**

| | Min | Max | Units |
|---|---|---|---|
| tTCPCfilter | 4 | 500 | µs |

**Table 4-~~20~~4-~~20~~. CC_STATUS Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7..6 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B5 | Looking4Connection | 0b: TCPC is not actively looking for a connection. A transition from '1' to '0' indicates a potential connection has been found.<br>1b: TCPC is looking for a connection (toggling as a DRP or looking for a connection as Sink/Source only condition) |
| B4 | ConnectResult | 0b: the TCPC is presenting Rp<br>1b: the TCPC is presenting Rd |
| B3..2 | CC2 State | If (ROLE_CONTROL.CC2=Rp) or (ConnectResult=0)<br>00b: SRC.Open (Open, Rp)<br>01b: SRC.Ra (below maximum vRa)<br>10b: SRC.Rd (within the vRd range)<br>11b: reserved<br><br>If (ROLE_CONTROL.CC2=Rd) or (ConnectResult=1)<br>00b: SNK.Open (Below maximum vRa)<br>01b: SNK.Default (Above minimum vRd-Connect)<br>10b: SNK.Power1.5 (Above minimum vRd-Connect) Detects Rp 1.5A<br>11b: SNK.Power3.0 (Above minimum vRd-Connect) Detects Rp 3.0A<br><br>If ROLE_CONTROL.CC2=Ra, this field is set to 00b<br>If ROLE_CONTROL.CC2=Open, this field is set to 00b<br><br>This field always returns 00b if (Looking4Connection=1) or (POWER_CONTROL.EnableVconn=1 and TCPC_CONTROL.PlugOrientation=0). Otherwise, the returned value depends upon ROLE_CONTROL.CC2. |
| B1..0 | CC1 State | If (ROLE_CONTROL.CC1 = Rp) or (ConnectResult=0)<br>00b: SRC.Open (Open, Rp)<br>01b: SRC.Ra (below maximum vRa)<br>10b: SRC.Rd (within the vRd range)<br>11b: reserved<br><br>If (ROLE_CONTROL.CC1 = Rd) or ConnectResult=1)<br>00b: SNK.Open (Below maximum vRa)<br>01b: SNK.Default (Above minimum vRd-Connect)<br>10b: SNK.Power1.5 (Above minimum vRd-Connect) Detects Rp-1.5A<br>11b: SNK.Power3.0 (Above minimum vRd-Connect) Detects Rp-3.0A<br><br>If ROLE_CONTROL.CC1=Ra, this field is set to 00b<br>If ROLE_CONTROL.CC1=Open, this field is set to 00b<br><br>This field always returns 00b if (Looking4Connection=1) or (POWER_CONTROL.EnableVconn=1 and TCPC_CONTROL.PlugOrientation=1). Otherwise, the returned value depends upon ROLE_CONTROL.CC1. |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.6.2 POWER_STATUS (Required)

The TCPM reads this register upon detecting an Alert# and reading the ALERT.PortPowerStatus Register set to 1. The TCPC indicates the current Power Status in this register.

The TCPM operating as a Sink at vSafe5V (with or without a ***USB PD~~USB PD~~*** Contract) shall detect VBUS presence and removal by reading the VBUSPresent bit.

Formatted: bookmark Char, Font: Not Bold

The TCPM shall check the state of the TCPC Initialization Status bit when it starts or resets. The TCPM shall not start normal operation until the TCPC Initialization Status bit is cleared. The TCPC shall set the TCPC Initialization Status bit to zero when initialization or reset is complete and all registers are valid.

**Table 4-~~21~~14-~~21~~. POWER_STATUS Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7 | Debug Accessory Connected | 0b: No Debug Accessory connected (default)<br>1b: Debug Accessory connected<br>Reflects the state of the DebugAccessoryConnected# output if supported<br>Required (Register is required but output is not required) |
| B6 | TCPC Initialization Status | 0b: The TCPC has completed initialization and all registers are valid<br>1b: The TCPC is still performing internal initialization and the only registers that are guaranteed to return the correct values are 00h..0Fh<br>Required |
| B5 | Sourcing High Voltage | 0b: vSafe5V<br>1b: High Voltage<br>This does not control the path, just provides a monitor of the status. Assert as long as supplying voltage greater than vSafe5V.<br>Required if voltage higher than vSafe5V can be sourced |
| B4 | Sourcing VBUS | 0b: Sourcing VBUS is disabled<br>1b: Sourcing VBUS is enabled<br>This does not control the path, just provides a monitor of the status.<br>Required |
| B3 | VBUS Present Detection Enabled | 0b: VBUS Present Detection Disabled<br>1b: VBUS Present Detection Enabled (default)<br>Indicates if the TCPC is monitoring for VBUS Present or if the circuit has been powered off<br>Required |
| B2 | VBUS Present | 0b: VBUS Disconnected<br>1b: VBUS Connected<br>The TCPC shall report VBUS present when TCPC detects VBUS rises above 4V. The TCPC shall report VBUS is not present when TCPC detects VBUS falls below 3.5V. The TCPC may report VBUS is not present if VBUS is between 3.5V and 4V.<br>Required |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

| Bit(s) | Name | Description |
|---|---|---|
| B1 | VCONN Present | 0b: VCONN is not present<br>1b: This bit is asserted when VCONN present CC1 or CC2. Threshold is fixed at 2.4V<br>If POWER_CONTROL.EnableVCONN is disabled VCONN Present should be set to 0b.<br>Required |
| B0 | Sinking VBUS | 0b: Sink is Disconnected (Default and if not supported)<br>1b: TCPC is sinking VBUS to the system load<br>Required |

#### 4.4.6.3 FAULT_STATUS (Required)

The TCPM reads this register upon detecting an Alert# and reading the ALERT.Fault bit set to 1. The TCPC indicates the current fault status in this register.

The TCPC indicates a Fault status change has occurred by presenting a logical 1 in the corresponding bit position in this register, presenting a logical 1 to the ALERT.Fault bit, and asserting the Alert# pin if the corresponding fault bit in FAULT_STATUS_MASK is 1 and ALERT_MASK.Fault is 1. The TCPM clears the FAULT bit by writing a logical 1 to the respective FAULT bit position and then writing a logical 1 to the ALERT.Fault bit after all bits in FAULT_STATUS have been cleared. The TCPM can clear any number of FAULT bits in a single write by setting multiple bits to logical 1 and the rest of the bits in the register to logical 0. The TCPM writing a logical 0 to any FAULT bit has no effect and therefore does not cause those FAULT bits to be set or cleared.

**Table 4-22~~4-22~~. FAULT_STATUS Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7 | AllRegistersResetToDefault~~Reserved~~ | This bit is asserted when the TCPC resets all registers to their default value. This happens at initial power up or if an unexpected power reset occurs.~~Shall be set to zero by sender and ignored by receiver~~ |
| B6 | Force Off VBUS<br>(Source or Sink) | 0b: No Fault Detected, no action (default and not supported)<br>1b: VBUS Source/Sink has been forced off due to external fault<br>The TCPC has disconnected VBUS due to STANDARD_INPUT.ForceOffVbus.<br>Required if STANDARD_INPUT_CAPABILITIES_1.ForceOffVbus = 1b |
| B5 | Auto Discharge Failed | 0b: No discharge failure<br>1b: Discharge commanded by the TCPM failed<br>If POWER_CONTROL.AutoDischargeDisconnect is set, the TCPC shall report discharge fails if VBUS is not below vSafe0V within tSafe0V.<br>Required |
| B4 | Force Discharge Failed | 0b: No discharge failure<br>1b: Discharge commanded by the TCPM failed<br>If POWER_CONTROL.ForceDischarge is set, the TCPC shall report a discharge fails if VBUS is not below vSafe0V within tSafe0V.<br>Required if DEVICE_CAPABILITIES_1.ForceDischarge =1b |

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

| Bit(s) | Name | Description |
|---|---|---|
| B3 | Internal or External OCP<br>VBUS Over Current Protection Fault | 0b: Not in an over-current protection state<br>1b: Over-current fault latched<br>Required if DEVICE_CAPABILIITES_1.VBUSOCPReporting = 1b |
| B2 | Internal or External OVP<br>VBUS Over Voltage Protection Fault | 0b: Not in an over-voltage protection state<br>1b: Over-voltage fault latched.<br>Required if DEVICE_CAPABILIITES_1.VBUSOVPReporting = 1b |
| B1 | VCONN Over Current Fault | 0b: No Fault detected<br>1b: Over current VCONN fault latched<br>Required if DEVICE_CAPABILITIES_2.VCONNOvercurrentFaultCapable = 1b |
| B0 | I2C Interface Error | 0b: No Error<br>1b: I2C error has occurred.<br>A TRANSMIT has been sent with an empty TRANSMIT_BUFFER. May be asserted if a non-zero value has been written to a reserved bit in a register.<br>Required |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.7 COMMAND (Required)

The Command is issued by the TCPM. The Command is cleared by the TCPC after being acted upon.

The TCPM shall issue COMMAND.Look4Connection to enable the TCPC to autonomously toggle the Rp/Rd starting with the Rp. The initial Rp or Rd for toggling is determined by ROLE_CONTROL.CC1 and ROLE_CONTROL.CC2. If ROLE_CONTROL.CC1 and ROLE_CONTROL.CC2 are not the same value, the COMMAND.Look4Connection shall have no effect.

The TCPM shall issue COMMAND.Look4Connection to enable the TCPC to restart Connection Detection in cases where the role is fixed as Source or Sink, ROLE_CONTROL.DRP = 0b. An example of this is when a potential connection as a Source occurred but was further debounced by the TCPM to find the Sink disconnected. In this case a Source Only or DRP should go back to its Unattached.Src state. This would result in ROLE_CONTROL staying the same.

COMMAND.I2CIdle is used to put the I2C interface into the idle state. When the TCPC receives COMMAND.I2CIdle, the TCPC may generate a bit-level Not Acknowledge signal (a NAK where SDA remains HIGH during the ninth clock pulse) to its own slave address or any I2C commands.

The TCPM may send the COMMAND.WakeI2C as a throw away command to wake the I2C interface. The COMMAND.WakeI2C requires no action by the TCPC other than to wake the I2C device interface in the TCPC.

COMMAND.I2CIdle is decoupled from other Alert status detection mechanisms (such as CC_STATUS, POWER_STATUS, RECEIVE_DETECT, etc). For example, writing COMMAND. I2CIdle has no effect on ALERT.CcStatus, or the CC_STATUS register behavior. CC_STATUS detection may be turned off by writing to the ROLE_CONTROL register, but its behavior is not affected by the COMMAND.I2CIdle.

The TCPM shall issue COMMAND.SourceVbusHighVoltage to enable the TCPC to transition the Vbus source to a higher voltage level. The target voltage level for COMMAND.SourceVbusHighVoltage is set in a vendor defined manner. The TCPM may need to send vendor defined commands before sending COMMAND.SourceVbusHighVoltage. The steps of transitioning to source a higher voltage than vSafev5V over Vbus may be as follow:

Step 0: TCPC supplies vSafe5V over Vbus
Step 1: TCPM issues vendor defined commands to set the target voltage level of COMMAND.SourceVbusHighVoltage
Step 2: TCPM issues COMMAND.SourceVbusHighVoltage
Step 3: TCPC starts the operation of transitioning to the target voltage level. TCPC shall control the voltage transitioning and meet the power supply requirements in the ***USB PDUSB PD*** specification. For TCPC that directly integrates the power switches, the TCPC shall meet the power supply requirements in the USB PD specification when the TCPC transitions VBUS voltage.

If the TCPM has a new target voltage level for COMMAND.SourceVbusHighVoltage, go to Step 1. The TCPM does not have to go back to vSafe5V and then to a different voltage. It may go directly to the new voltage by setting a new voltage level and then issuing the COMMAND.SourceVbusHighVoltage.

Figure 4-8 and Figure 4-9 indicate the flow from vSafe5V to and from high voltage respectively.

**Formatted:** bookmark Char, Font: Not Bold

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

**Table 4-~~23~~24-~~23~~. COMMAND Register Definition**

| Bit(s) | Name | Register Setting | Description |
|---|---|---|---|
| B7..0 | Command | 0001 0001b | WakeI2C (no action is taken other than to wake the I2C interface). |
| | | 0010 0010b | DisableVbusDetect. Disable Vbus present detection. The TCPC shall ignore this command and assert the FAULT_STATUS.I2CInterfaceError if it has sourcing or sinking power over Vbus enabled. |
| | | 0011 0011b | EnableVbusDetect. Enable Vbus present detection. |
| | | 0100 0100b | DisableSinkVbus. Disable sinking power over Vbus. This COMMAND does not disable POWER_STATUS.VBUSPresent detection.<br>The TCPC shall clear FAULT_STATUS.InternalorExternalOCP and FAULT_STATUS.InternalorExternalOVP. |
| | | 0101 0101b | SinkVbus. Enable sinking power over Vbus and enable Vbus present detection. The TCPC shall ignore this command and assert the FAULT_STATUS.I2CInterfaceError if it has sourcing power over Vbus enabled. |
| | | 0110 0110b | DisableSourceVbus. Disable sourcing power over Vbus. The TCPC shall stop reporting FAULT_STATUS. Internal or External OCP or OVP Faults. This COMMAND does not disable POWER_STATUS.VBUSPresent detection. |
| | | 0111 0111b | SourceVbusDefaultVoltage. Enable sourcing vSafe5V over Vbus and enable Vbus present detection. Source shall transition to vSafe5V if at a high voltage. The TCPC shall ignore this command and assert the FAULT_STATUS.I2CInterfaceError if it has sinking power over Vbus enabled. |
| | | 1000 1000b | SourceVbusHighVoltage. Execute sourcing high voltage over Vbus. The TCPC shall ignore this command and assert the FAULT_STATUS.I2CInterfaceError if it is currently sinking voltage from Vbus or does not have ability to source voltages higher than vSafe5V. The TCPC shall ignore this command and assert the FAULT_STATUS.I2CInterfaceError if not already sourcing vSafe5V.<br>The actual voltage to be sourced may be set in a vendor defined manner. The TCPM may need to send vendor defined commands before sending this COMMAND. |
| | | 1001 1001b | Look4Connection. Start DRP Toggling if ROLE_CONTROL.DRP=1b. If ROLE_CONTROL.CC1/CC2 = 01b start with Rp, if ROLE_CONTROL.CC1/CC2 =10b start with Rd.<br>If ROLE_CONTROL.CC1/CC2 are not both 01b or 10b, then do not start toggling.<br>The TCPM shall issue COMMAND.Look4Connection to enable the TCPC to restart Connection Detection in cases where the ROLE_CONTROL contents will not change. An example of this is when a potential connection as a Source occurred but was further debounced by the TCPM to find the Sink disconnected. In this case a Source Only or DRP should go back to its Unattached.Src state. This would result in ROLE_CONTROL staying the same.<br>TCPC to MAINTAIN_STATE in ~~Figure 4-11~~Figure 4-11 |
| | | 1010 1010b | RxOneMore. Configure the receiver to automatically clear the RECEIVE_DETECT register after sending the next GoodCRC. This is used to shutdown reception of packets at a known point regardless of packet separation or the depth of the receive FIFO in the TCPC. |
| | | 1100 1100b<br>1101 1101b<br>1110 1110b | Reserved. No Action |
| | | 1111 1111b | I2C Idle |

Note: All other values are reserved; shall be ignored by the receiver

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

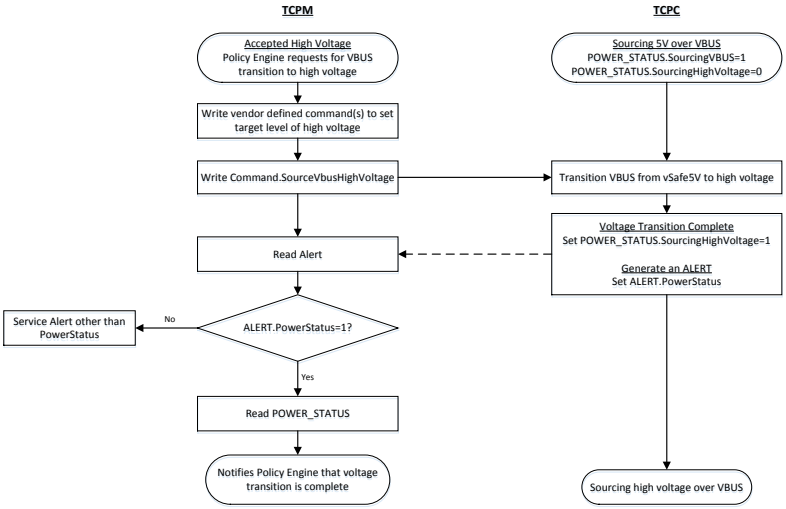



**Figure 4-~~8~~4-~~8~~. Transition from vSafe5V to High Voltage**

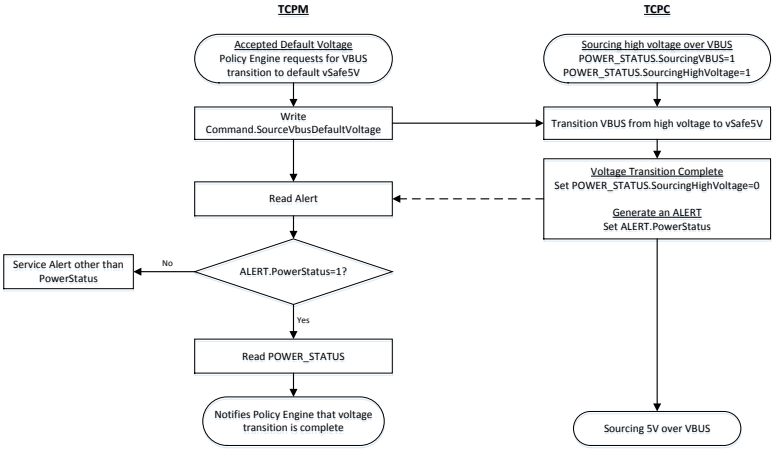



**Figure 4-~~9~~4-~~9~~. Transition from High Voltage to vSafe5V**

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.8 Capability Registers

This set of registers is used to communicate the capabilities of the TCPC to the TCPM. The TCPM reads these registers.

##### 4.4.8.1 DEVICE_CAPABILITIES (Required)

This register is in the nonvolatile memory of the TCPC. This register describes features supported by the TCPC.

**Table 4-~~24~~4-~~24~~. DEVICE_CAPABILITIES_1 Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B15 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B14 | VBUS OCP Reporting | 0b: VBUS OCP is not reported by the TCPC<br>1b: VBUS OCP is reported by the TCPC<br>Support for both FAULT_STATUS.InternalorExternalOCP and FAULT_CONTROL. InternalorExternalOCP implemented |
| B13 | VBUS OVP Reporting | 0b: VBUS OVP is not reported by the TCPC<br>1b: VBUS OVP is reported by the TCPC<br>Support for both FAULT_STATUS.InternalorExternalOVP and FAULT_CONTROL. InternalorExternalOVP implemented |
| B12 | Bleed Discharge | 0b: No Bleed Discharge implemented in TCPC<br>1b: Bleed Discharge is implemented in the TCPC<br>Support for POWER_CONTROL.EnableBleedDischarge implemented |
| B11 | Force Discharge | 0b: No Force Discharge implemented in TCPC<br>1b: Force Discharge is implemented in the TCPC<br>Support for POWER_CONTROL.ForceDischarge, FAULT_STATUS.VbusDischargeFail, FAULT_STATUS.VBUSDischargeFaultDetectionTimer, and VBUS_STOP_DISCHARGE_THRESHOLD implemented |
| B10 | VBUS Measurement and Alarm Capable | 0b: No VBUS voltage measurement nor VBUS Alarms<br>1b; VBUS voltage measurement and VBUS Alarms<br>Support for VBUS_VOLTAGE, VBUS_VOLTAGE_ALARM_HI_CFG, VBUS_VOLTAGE_ALARM_LO_CFG implemented |
| B9..8 | Source Resistor Supported | 00b: Rp default only<br>01b: Rp 1.5A and default<br>10b: Rp 3.0A, 1.5A, and default<br>11b: Reserved<br>Rp values which may be configured by the TCPM via the ROLE_CONTROL register |
| B7..5 | Roles Supported | 000b: USB Type-C Port Manager can configure the Port as Source only or Sink only (not DRP)<br>001b: Source only<br>010b: Sink only<br>011b: Sink with accessory support<br>100b: DRP only<br>101b: Source, Sink, DRP, Adapter/Cable all supported<br>110b: Source, Sink, DRP<br>111b: Not valid |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

| Bit(s) | Name | Description |
|---|---|---|
| B4 | SOP'_DBG/SOP"_DBG Support | 0b: All SOP* except SOP'_DBG/SOP"_DBG<br>1b: All SOP* messages are supported<br>Configured in RECEIVE_DETECT and TRANSMIT |
| B3 | Source VCONN | 0b: TCPC is not capable of switching VCONN<br>1b: TCPC is capable of switching VCONN<br>Support for POWER_CONTROL.EnableVCONN and POWER_STATUS.VCONNPresent implemented |
| B2 | Sink VBUS | 0b: TCPC is not capable controlling the sink path to the system load<br>1b: TCPC is capable of controlling the sink path to the system load<br>Support for POWER_STATUS.SinkingVbus, COMMAND.SinkVbus, and COMMAND.DisableSinkVbus implemented |
| B1 | Source High Voltage VBUS | 0b: TCPC is not capable of controlling the source high voltage path to VBUS<br>1b: TCPC is capable of controlling the source high voltage path to VBUS<br>Support for POWER_STATUS.SourcingHighVoltage and COMMAND.SourceVbusHighVoltage implemented |
| B0 | Source VBUS | 0b: TCPC is not capable of controlling the source path to VBUS<br>1b: TCPC is capable of controlling the source path to VBUS<br>Support for POWER_STATUS.SourcingVbus, COMMAND.SourceVbusDefaultVoltage, COMMAND.DisableSourceVbus, COMMAND.EnableVbusDetect and COMMAND.DisableVbusDetect implemented |

**Table 4-~~25~~25~~4-25~~. DEVICE_CAPABILITIES_2 Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B15..~~9~~8 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B8 | Watchdog Timer | 0b: TCPC_CONTROL.Enable Watchdog Timer not implemented<br>1b: TCPC_CONTROL.Enable Watchdog Timer implemented |
| B7 | Sink Disconnect Detection | 0b: VBUS_SINK_DISCONNECT_THRESHOLD not implemented (default: Use POWER_STATUS.VbusPresent=0b to indicate a Sink disconnect)<br>1b: VBUS_SINK_DISCONNECT_THRESHOLD implemented |
| B6 | Stop Discharge Threshold | 0b: VBUS_STOP_DISCHARGE_THRESHOLD not implemented (default)<br>1b: VBUS_STOP_DISCHARGE_THRESHOLD implemented |

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

| Bit(s) | Name | Description |
|---|---|---|
| B5..4 | VBUS Voltage Alarm LSB | 00: TCPC has 25mV LSB for its voltage alarm and uses all 10 bits in VBUS_VOLTAGE_ALARM_HI_CFG and VBUS_VOLTAGE_ALARM_LO_CFG.<br>01: TCPC has 50mV LSB for its voltage alarm and uses only 9 bits. VBUS_VOLTAGE_ALARM_HI_CFG[0] and VBUS_VOLTAGE_ALARM_LO_CFG[0] are ignored by TCPC.<br>10: TCPC has 100mV LSB for its voltage alarm and uses only 8 bits. VBUS_VOLTAGE_ALARM_HI_CFG[1:0] and VBUS_VOLTAGE_ALARM_LO_CFG[1:0] are ignored by TCPC.<br>11: reserved<br>Support for VBUS_VOLTAGE_ALARM_LO_CFG and VBUS_VOLTAGE_ALARM_HI implemented |
| B3..1 | VCONN Power Supported | 000b: 1.0W<br>001b: 1.5W<br>010b: 2.0W<br>011b: 3W<br>100b: 4W<br>101b: 5W<br>110b: 6W<br>111b: External |
| B0 | VCONN Overcurrent Fault Capable | 0b: TCPC is not capable of detecting a $V_{conn}$ fault<br>1b: TCPC is capable of detecting a $V_{conn}$ fault<br>Support for FAULT_STATUS.VCONNOverCurrentFault and FAULT_CONTROL.VCONNOverCurrentFault implemented |

#### 4.4.8.2 STANDARD_INPUT_CAPABILITIES (Required)

This register is in the nonvolatile memory of the TCPC. This register describes the required Standard Inputs and their capability.

**Table 4-~~26~~4-~~26~~. STANDARD_INPUT_CAPABILITIES Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7..3 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B2 | VBUS External Over Voltage Fault | 0b: Not present in TCPC<br>1b: Present in TCPC |
| B1 | VBUS External Over Current Fault | 0b: Not present in TCPC<br>1b: Present in TCPC |
| B0 | Force Off VBUS (Source or Sink) | 0b: Not present in TCPC<br>1b: Present in TCPC |

#### 4.4.8.3 STANDARD_OUTPUT_CAPABILITIES (Required)

This register is in the nonvolatile memory of the TCPC. This register describes the required Standard Outputs and their capability. The Standard Outputs are push/pull and referenced to a VDDIO which may be the same or different than the VDD supply voltage for the I2C interface.

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

**Table 4-~~27~~4-~~27~~. STANDARD_OUTPUT_CAPABILITIES Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7 | Reserved | Shall be set to zero |
| B6 | Debug Accessory Indicator | 0b: Not present in TCPC<br>1b: Present in TCPC |
| B5 | VBUS Present Monitor | 0b: Not present in TCPC<br>1b: Present in TCPC |
| B4 | Audio Adapter Accessory Indicator | 0b: Not present in TCPC<br>1b: Present in TCPC |
| B3 | Active Cable Indicator | 0b: Not present in TCPC<br>1b: Present in TCPC |
| B2 | MUX Configuration Control | 0b: Not present in TCPC<br>1b: Present in TCPC |
| B1 | Connection Present | 0b: No Connection<br>1b: Connection<br>Controlled by the TCPM. |
| B0 | Connector Orientation | 0b: Not present in TCPC<br>1b: Present in TCPC |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.9 MESSAGE_HEADER_INFO (Required)

The TCPC shall set this register at power on per Table 4-15. The TCPM may overwrite this register after TCPC initialization is complete.

On attach and after implementing the tCCDebounce, the TCPM shall update the MESSAGE_HEADER_INFO Register first before writing to the RECEIVE_DETECT register.

The TCPC reads from this register to generate the Message header for the GoodCRC.

**Table 4-~~28~~4-~~28~~. MESSAGE_HEADER_INFO Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7..5 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B4 | Cable Plug | 0b: Message originated from Source, Sink, or DRP<br>1b: Message originated from a Cable Plug |
| B3 | Data Role | 0b: UFP<br>1b: DFP |
| B2..1 | ~~*USB PDUSB PD*~~ Specification Revision | 00b: Revision 1.0<br>01b: Revision 2.0<br>10b – 11b: Reserved |
| B0 | Power Role | 0b: Sink<br>1b: Source |

Formatted: bookmark Char, Font: 8 pt, Not Bold

#### 4.4.10 RECEIVE_DETECT (Required)

Set by TCPM, cleared by TCPM (and/or TCPC in some instances).

The TCPM notifies the TCPC of the message type and/or signaling types to be detected. The TCPM should not set any bits in this register until it is able to respond. The TCPC responds to the enabled message type with a GoodCRC if it is a SOP* message, except in the case of a GoodCRC message.

The initial value of RECEIVE_DETECT shall be all zeroes. Registers must be written to be enabled.

On Hard Reset reception, the TCPC shall set the RECEIVE_DETECT bits to zero to disable automatic transmission of GoodCRC. On detection of a Disconnect, the TCPC shall set the RECEIVE_DETECT bits all to zero to disable automatic transmission of GoodCRC. The TCPC shall set the RECEIVE_DETECT bits to zero to disable automatic transmission of GoodCRC when RECEIVE_DETECT.CableReset is set and a Cable Reset is received. The TCPC shall be capable of receiving a CableReset if DEVICE_CAPABILITIES_1.RolesSupported =101, the TCPC may be capable of receiving a CableReset DEVICE_CAPABILITIES_1.RolesSupported != 101.

Refer to Section 4.7.3 for more detail.

**Table 4-~~29~~4-~~29~~. RECEIVE_DETECT Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7 | Reserved | Shall be set to zero by sender and ignored by receiver |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

| Bit(s) | Name | Description |
|---|---|---|
| B6 | Enable Cable Reset | 0b: TCPC does not detect Cable Reset signaling (default)<br>1b: TCPC detects Cable Reset signaling |
| B5 | Enable Hard Reset | 0b: TCPC does not detect Hard Reset signaling (default)<br>1b: TCPC detects Hard Reset signaling |
| B4 | Enable SOP_DBG'' message | 0b: TCPC does not detect SOP_DBG'' message (default)<br>1b: TCPC detects SOP_DBG'' message |
| B3 | Enable SOP_DBG' message | 0b: TCPC does not detect SOP_DBG' message (default)<br>1b: TCPC detects SOP_DBG' message |
| B2 | Enable SOP'' message | 0b: TCPC does not detect SOP'' message (default)<br>1b: TCPC detects SOP'' message |
| B1 | Enable SOP' message | 0b: TCPC does not detect SOP' message (default)<br>1b: TCPC detects SOP' message |
| B0 | Enable SOP message | 0b: TCPC does not detect SOP message (default)<br>1b: TCPC detects SOP message |

#### 4.4.11 RECEIVE_BUFFER (Required)

This register indicates the status of the received SOP* message buffer in 30h..4Fh registers. This register shall be read by the TCPM when the TCPC indicates a SOP* message was received in the Alert Status registers. The TCPM reads the RECEIVE_BYTE_COUNT to determine the number of bytes in the RX_BUFFER_DATA_OBJECT and the RX_BUF_FRAME_TYPE to determine the type of message.

The TCPM then reads the information in the RX_BUF_HEADER and the RX_BUFFER_DATA_OBJECT. The TCPC shall set the RECEIVE_BYTE_COUNT to 0 after the interrupt has been cleared. See Section 4.7.3, and 4.7.4, and 4.7.5 for information on receiving SOP* messages, and Hard Reset, and Cable Reset messages respectively.

Upon detection of a Disconnect, the TCPC shall set the RECEIVE_BYTE_COUNT to zero. The TCPM may power down PD messaging per Section 4.8.2.

RECEIVE_BUFFER is read only.

**Table 4-304-30. RECEIVE_BYTE_COUNT Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7..0 | RECEIVE_BYTE_COUNT | Indicates number of bytes in this register that are not stale. The TCPM should read the first RECEIVE_BYTE_COUNT bytes in this register. This is the number of bytes in the RX_BUFFER_DATA_OBJECTS plus three (for the RX_BUF_FRAME_TYPE and RX_BUF_HEADER). |

**Table 4-314-31. RX_BUF_FRAME_TYPE Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7..3 | Reserved | Shall be set to zero by sender and ignored by receiver |

Copyright © 2015-2016 USB 3.0 Promoter Group. All rights reserved.

| Bit(s) | Name | Description |
|---|---|---|
| B2..0 | Received SOP* Message | 000b: Received SOP<br>001b: Received SOP'<br>010b: Received SOP''<br>011b: Received SOP_DBG'<br>100b: Received SOP_DBG''<br>110b: Received Cable Reset<br>All others are reserved. |

**Table 4-~~32~~4-~~32~~. RX_BUF_HEADER Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B15..8 | RX_BUF_HEADER_BYTE_1 | Byte 1 (bits 15..8) of ~~*USB PD*~~*USB PD* message header |
| B7..0 | RX_BUF_HEADER_BYTE_0 | Byte 0 (bits 7..0) of ~~*USB PD*~~*USB PD* message header |

**Formatted:** bookmark Char, Font: 8 pt, Not Bold

**Formatted:** bookmark Char, Font: 8 pt, Not Bold

**Table 4-~~33~~4-~~33~~. RX_BUFFER_DATA_OBJECTS Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B223..8 | RX_BUF_OBJn_BYTE_m | Byte m of nth data object |
| B7..0 | RX_BUF_OBJ1_BYTE_0 | Byte 0 (bits 7..0) of 1st data object |

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.12 TRANSMIT (Required)

The TCPM writes to this register to transmit a SOP* message where the SOP* message payload is in 51h..6Fh registers. The entire register shall be written at once and then sent. The TCPC shall clear the TRANSMIT register and TRANSMIT_BYTE_COUNT after executing the transmission regardless of success or failure.

If the TCPM writes to TRANSMIT requesting a transmission that is not Hard Reset, not Cable Reset, and not BIST Carrier Mode 2 (i.e. TRANSMIT.SOP*Message > 100b) and TRANSMIT_BYTE_COUNT is less than ~~3h~~2h, the TCPC shall generate a FAULT_STATUS.I2CInterfaceError.

The TCPM shall require no message retry when transmitting a Hard Reset, Cable Reset, or BIST Carrier Mode 2 signaling. The TCPC shall ignore the Retry Counter bits (B5..4) when transmitting a Hard Reset, Cable Reset, or BIST Carrier Mode signaling.

The tCableMessage timer (SOP' and SOP") shall be in the TCPM.

The TCPC is not allowed to NAK this register.

If the TCPM writes a Hard Reset command to this register while a transmission is in progress, a Hard Reset signal shall be sent as soon as possible (interrupting the outgoing message according to the PD specifications, or transmitting it after the GoodCRC reply to previous command has been received or CRCReceiveTimer has expired).

The TCPM shall not write to the TRANSMIT register to request a transmission other than Hard Reset while the TCPC is still processing a previous transmission (i.e. ALERT.TransmitSuccessful, TransmitFailed, or TransmitDiscarded have not yet been asserted since the last write to the TRANSMIT register). The TCPM shall clear the resulting alert from a prior TRANSMIT write before writing to the TRANSMIT register again for anything other than a Hard Reset. If a previous TRANSMIT request has not yet completed when TRANSMIT is written requesting a Hard Reset, the TCPC shall assert the Transmission Discarded bit in the ALERT register.

The TCPM shall not write to the TRANSMIT register to request a transmission other than Hard Reset until it has cleared all received message alerts. If the TCPM writes TRANSMIT when ALERT.ReceivedHardReset = 1 or ALERT.ReceivedSOP* = 1 the TCPC shall discard the transmit request and assert ALERT.TxMessageDiscarded.

The TCPC shall assert one and only one of ALERT.TransmitSuccessful, TransmitFailed, or TransmitDiscarded following a write of the TRANSMIT register when a non-Hard Reset is transmitted. The TCPC shall assert both ALERT.TransmitSuccessful and ALERT.TransmitFailed after it completes the sending of a Hard Reset. The TCPC shall clear the RECEIVE_DETECT and the RECEIVE_BYTE_COUNT register to disable the PD message passing when the TCPM writes the TRANSMIT register requesting a Hard Reset.

**Table 4-~~34~~4-~~34~~. TRANSMIT Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7..6 | Reserved | Shall be set to zero by sender and ignored by receiver |
| B5..4 | Retry Counter | 00b: No message retry is required<br>01b: Automatically retry message transmission once<br>10b: Automatically retry message transmission twice<br>11b: Automatically retry message transmission three times |
| B3 | Reserved | Shall be set to zero by sender, shall be ignored by receiver |

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

| Bit(s) | Name | Description |
|---|---|---|
| B2..0 | Transmit SOP* message | 000b: Transmit SOP<br>001b: Transmit SOP'<br>010b: Transmit SOP''<br>011b: Transmit SOP_DBG'<br>100b: Transmit SOP_DBG''<br>101b: Transmit Hard Reset<br>110b: Transmit Cable Reset<br>111b: Transmit BIST Carrier Mode 2 (TCPC shall exit the BIST mode no later than tBISTContMode max) |

#### 4.4.13 TRANSMIT_BUFFER (Required)

The TRANSMIT_BUFFER holds the TRANSMIT_BYTE_COUNT, the TX_BUF_HEADER, and the TX_BUFFER_DATA_OBJECTS (SOP* payload).

**Table 4-~~35~~4~~-35~~. TRANSMIT_BYTE_COUNT Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B7..0 | TRANSMIT_BYTE_COUNT | The number of bytes the TCPM will write<br>This is the number of bytes in the TX_BUFFER_DATA_OBJECTS plus two (for the TX_BUF_HEADER) |

**Table 4-~~36~~4~~-36~~. TX_BUF_HEADER Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B15..8 | TX_BUF_HEADER_BYTE_1 | Byte 1 (bits 15..8) of ~~*USB PD*~~*USB PD* message header |
| B7..0 | TX_BUF_HEADER_BYTE_0 | Byte 0 (bits 7..0) of ~~*USB PD*~~*USB PD* message header |

**Formatted:** bookmark Char, Font: 8 pt, Not Bold

**Formatted:** bookmark Char, Font: 8 pt, Not Bold

**Table 4-~~37~~4~~-37~~. TX_BUFFER_DATA_OBJECTS Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B223..8 | TX_BUF_OBJn_BYTE_m | Byte m of nth data object |
| B7..0 | TX_BUF_OBJ1_BYTE_0 | Byte 0 (bits 7..0) of 1st data object |

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.14 VBUS_VOLTAGE (Optional Normative)

The TCPM may read this register to determine the VBUS voltage measured on the Source or Sink at the ***~~USB Type-CUSB Type-C~~*** Connector. The TCPC shall maintain synchronization between the upper and lower 8 bits of the register.

Formatted: bookmark Char, Font: Not Bold

The implementation of VBUS sampling for VBUS_VOLTAGE reporting in TCPC is vendor specific. The TCPC datasheet should provide TCPM guidance for reading VBUS_VOLTAGE register, such as the averaging of reads and the interval between reads.

Required if voltage greater than vSafe5V is sourced or sinked. This register is required if it is reported as supported in the one of the DEVICE_CAPABILITES registers, Section 4.4.8.1.

**Table 4-~~38~~4-~~38~~. VBUS_VOLTAGE Register Definition**

| Bit(s) | Name | Description |
|---|---|---|
| B15..12 | Reserved | Shall be set to 0 |
| B11..10 | Scale Factor | 00: VBUS measurement not scaled.<br>01: VBUS measurement divided by 2<br>10: VBUS measurement divided by 4<br>11: reserved |
| B9..0 | VBUS voltage measurement | 10-bit measurement of (VBUS / Scale Factor)<br>TCPM multiplies this value by the scale factor to obtain the voltage measurement. All V~~v~~oltages ~~greater than or equal to 4V~~ shall meet +/-2% absolute value or +/- 50mV, whichever is greater. The LSB is 25mV. |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.15 Voltage Thresholds

##### 4.4.15.1 VBUS _SINK_DISCONNECT_THRESHOLD

This register is required by TCPCs which act as a Sink and are capable of receiving voltages greater than vSafe5V. Implementation of this register shall be defined in DEVICE_CAPABILITIES_2.SinkDisconnectDetection.

This register is optional normative for TCPCs acting as Source only. This register has no action for a Source.

TCPM writes to this register to set the threshold at which a Sink will start the Auto Discharge if POWER_CONTROL.AutoDischargeDisconnect = 1b.

**Table 4-394-39. VBUS_SINK_DISCONNECT_THRESHOLD Register Description**

| Bit(s) | Name | Description |
|---|---|---|
| B15..10 | Reserved | Shall be set to 0 |
| B9..0 | Voltage trip point | 10-bit for voltage threshold with 25mV LSB. (Default vSafe5V)<br>+/- 5% accuracy.<br>A value of B9:0=000h disables this threshold<br>The TCPC may ignore B0 or B1 & B0 depending upon DEVICE_CAPABILITIES_2.VbusVoltageAlarmLsb. |

##### 4.4.15.2 VBUS_STOP_DISCHARGE_THRESHOLD

This register is optional normative. It may be supported in one power role and not in the other (example: supported in Source power role and not in Sink power role).

This register is required by TCPCs which act as a Source and support POWER_CONTROL.ForceDischarge. The TCPM writes to this register to set the threshold at which a Source shall stop the forced discharge when POWER_CONTROL.ForceDischarge = 1b. TCPC acting as a Source shall always discharge to vSafe0V upon disconnect, Hard Reset, or Power Role Swap. A Source TCPC which does not support this register shall discharge Vbus to vSafe0V.

This register is optional normative for TCPCs which act as a Sink. This register has no action for a Sink. On a disconnect, a Source TCPC discharges VBUS to vSafe0V as described in Section 4.4.5.4.2. Implementation of this register shall be defined in DEVICE_CAPABILITIES_2.StopDischargeThreshold.

**Table 4-404-40. VBUS_STOP_DISCHARGE_THRESHOLD Register Description**

| Bit(s) | Name | Description |
|---|---|---|
| B15..10 | Reserved | Shall be set to 0 |
| B9..0 | Voltage trip point | 10-bit for voltage threshold with 25mV LSB. (Default vSafe0V)<br>+/- 5% accuracy.<br>The TCPC may ignore B0 or B1 & B0 depending upon DEVICE_CAPABILITIES_2.VbusVoltageAlarmLsb. |

Copyright © 2015-2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.15.3 Voltage Alarms (Optional Normative)

These registers are required if it is reported as supported in the one of the DEVICE_CAPABILITES registers, Section 4.4.8.1.

Voltage alarms are required by TCPCs with integrated power FETs which handle voltages higher than vSafe5V.

The TCPM can write to POWER_CONTROL.DisableVoltageAlarms = 1b to disable the voltage alarms. The TCPM writes to this registers to set the high voltage alarm level. The TCPC sets ALERT.VBUSVoltageAlarmHi to 1 when VBUS exceeds the over voltage level.

**Table 4-~~41~~4-~~41~~. VBUS_VOLTAGE_ALARM_HI_CFG Register Description**

| Bit(s) | Name | Description |
|---|---|---|
| B15..10 | Reserved | Shall be set to 0 |
| B9..0 | Voltage trip point | 10-bit for voltage threshold with 25mV LSB.<br>+/- 5% accuracy.<br>The TCPC may ignore B0 or B1 & B0 depending upon DEVICE_CAPABILITIES_2.VbusVoltageAlarmLsb. |

The TCPM writes to this registers to set the ~~high~~ low voltage alarm level. The TCPC sets ALERT.VBUSVoltageAlarmLo to 1 when VBUS drops below the under voltage level.

**Table 4-~~42~~4-~~42~~. VBUS_VOLTAGE_ALARM_LO_CFG Register Description**

| Bit(s) | Name | Description |
|---|---|---|
| B15..10 | Reserved | Shall be set to 0 |
| B9..0 | Voltage trip point | 10-bit for voltage threshold with 25mV LSB.<br>+/- 5% accuracy.<br>The TCPC may ignore B0 or B1 & B0 depending upon DEVICE_CAPABILITIES_2.VbusVoltageAlarmLsb. |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

#### 4.4.16 VENDOR_DEFINED Registers

The behavior of these registers is exclusively defined by the vendor. There is no defined behavior.

The TCPM should process/write Vendor Specific bits only if it recognizes the device and according to the specifications provided by that vendor.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### 4.5 STANDARD IO SIGNALS

This section defines the signaling on the Standard Inputs and Outputs.

#### 4.5.1 STANDARD INPUT SIGNALS (Optional Normative)

This register is required if it is reported as supported in the one of the DEVICE_CAPABILITES registers (Section 4.4.8.1). This section defines the Standard Input signals to the TCPC. The input signals provide updates to the FAULT_STATUS register (Section 4.4.6.3). The TCPC shall set the FAULT_STATUS register based on the input level at the pin.

**Table 4-~~43~~4-~~43~~. Standard Input Signals**

| Inputs | Type |
|---|---|
| VBUS External Over Current Fault | Low: Set STANDARD_INPUT Register to 0. No Over Current Fault<br>High: Set STANDARD_INPUT Register to 1. Over Current Fault present<br>Reported in FAULT_STATUS.InternalExternalOCP |
| VBUS External Over Voltage Fault | Low: Set STANDARD_INPUT Register to 0. No Over Voltage Fault.<br>High: Set STANDARD_INPUT Register to 1. Over Voltage Fault present.<br>Reported in FAULT_STATUS.InternalExternalOVP |
| Force Off VBUS (Source or Sink) | Low: Set STANDARD_INPUT Register to 0. Do not force VBUS off.<br>High: Set STANDARD_INPUT Register to 1. Force VBUS Off.<br>Reported in FAULT_STATUS.ForceOffVBUS |

#### 4.5.2 STANDARD OUTPUT SIGNALS (Optional Normative except Alert#)

This register is required if it is reported as supported in the one of the DEVICE_CAPABILITES registers (Section 4.4.8.1). This section defines the Standard Output signals from the TCPC. The output signals may or may not be controlled by the TCPM. Behavior is defined in Table 4-44.

Outputs may be Push/Pull or Open Drain. Refer to the TCPC datasheet for definition. Outputs which are Push/Pull are referenced to VDDIO and may be the same or different than the VDD supply voltage for the I2C interface. Outputs where noted are tri-stated on disconnect.

**Table 4-~~44~~4-~~44~~. Standard Output Signals**

| Output | Type |
|---|---|
| Alert# | Low: Alert. The TCPC is indicating an Alert Status change has occurred. The TCPM shall read the ALERT Register to determine what event triggered the Alert.<br>High: No Alert<br>Open Drain |
| Debug Accessory Connected# | High: No Debug Accessory connected<br>Low: Debug Accessory connected<br>Push/Pull or Open Drain |
| VBUS Present# | Low: VBUS is present<br>High: VBUS is not present<br>Push/Pull or Open Drain |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

| Output | Type |
|---|---|
| Audio Accessory Connected# | High: No Audio Accessory connected<br>Low: Audio Accessory connected<br>Push/Pull or Open Drain |
| Active Cable Connected | High: Active Cable connected<br>Low: No Active Cable connected<br>Push/Pull or Open Drain |
| MUX Control 1 | Low: No DP Alternate Mode<br>High: DP Alternate Mode<br>Push/Pull or Open Drain |
| MUX Control 0 | Low: No connection or No USB Connection<br>High: USB Connection<br>Push/Pull or Open Drain |
| Connection Present | Low: No Connection<br>High: Connection<br>Push/Pull or Open Drain |
| Connector Orientation | Low: Normal (default)<br>High: Flipped<br>Push/Pull or Open Drain |

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

## 4.6 TCPC Connection State ~~Machine~~ Diagrams and Flows

Refer to section 3.5 for the structure of the state diagram figures.

Formatted: PARAGRAPH

~~4.6~~

This section defines the behavior of a TCPC when using the DRP functionality.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

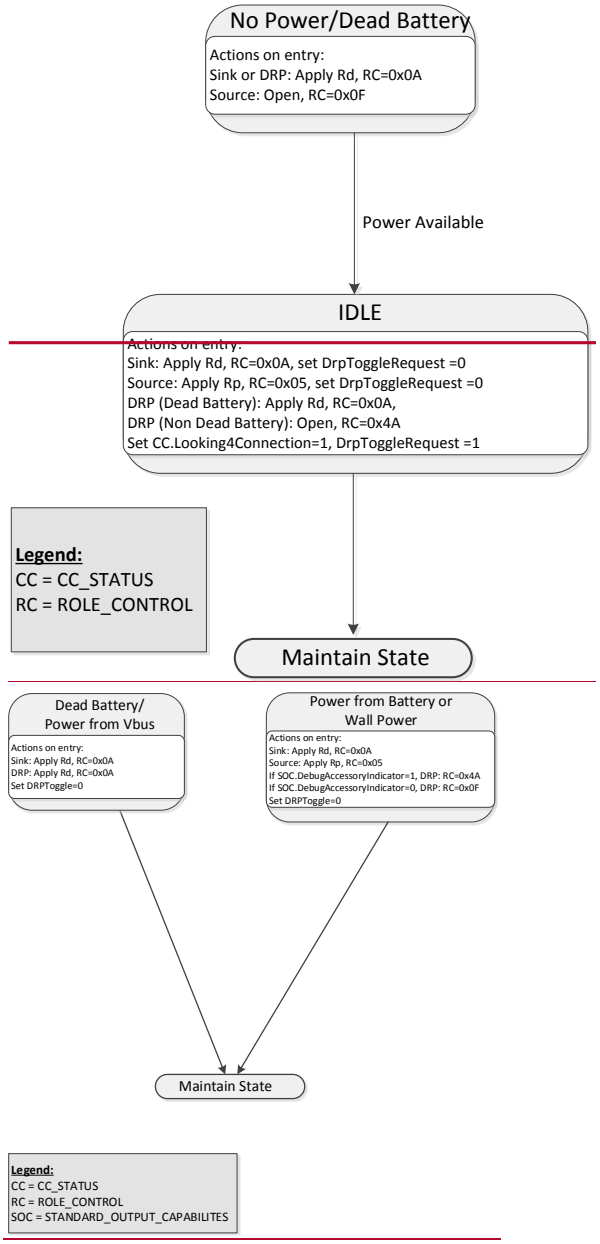
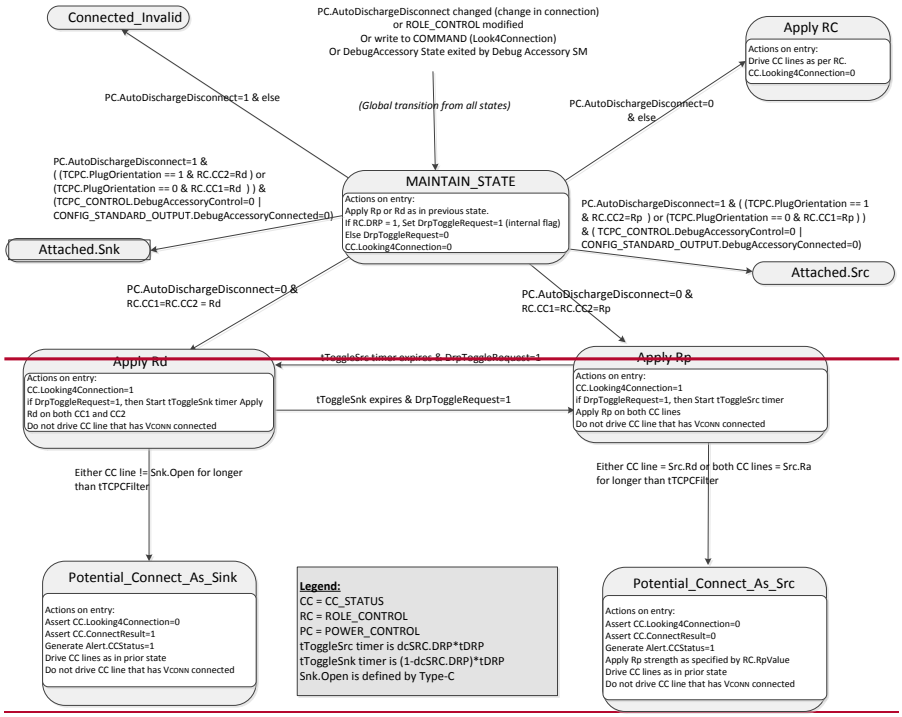



**Figure 4-~~10~~4-~~10~~. TCPC Power-On State ~~Machine~~Diagram**

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.




Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

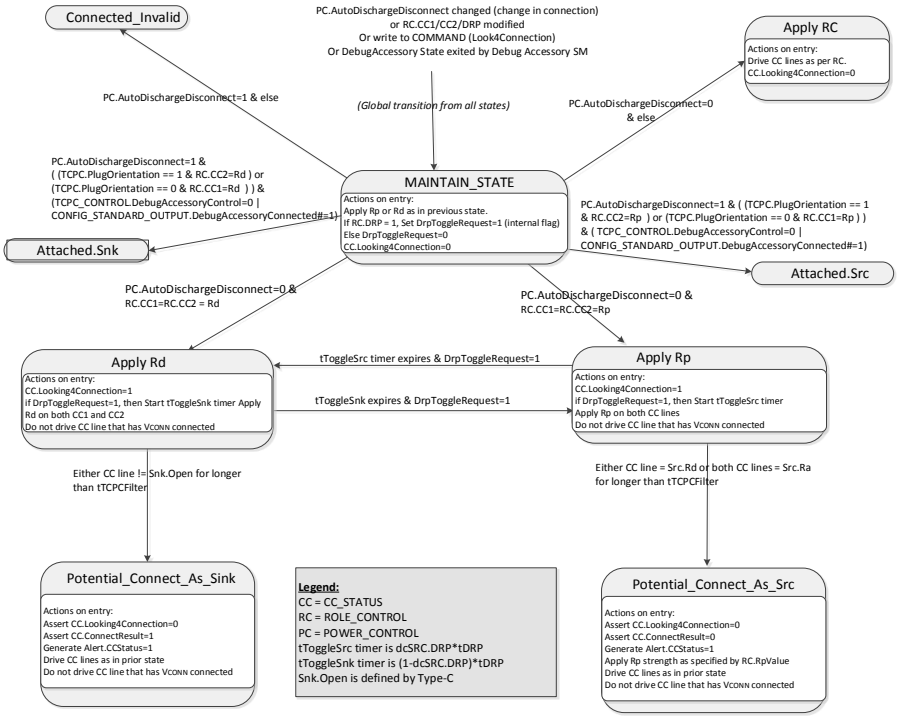



**Figure 4-~~11~~4-~~11~~. TCPC State ~~Machine~~ Diagram before a Connection**

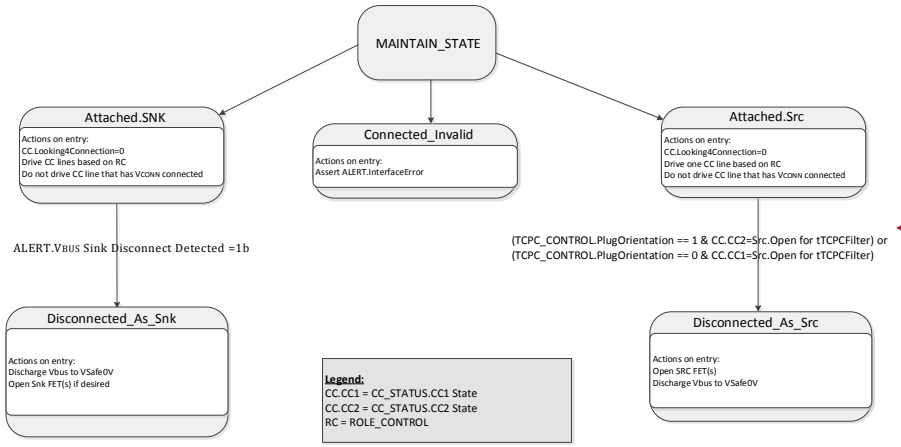



**Figure 4-12. TCPC State Diagram after a Connection**

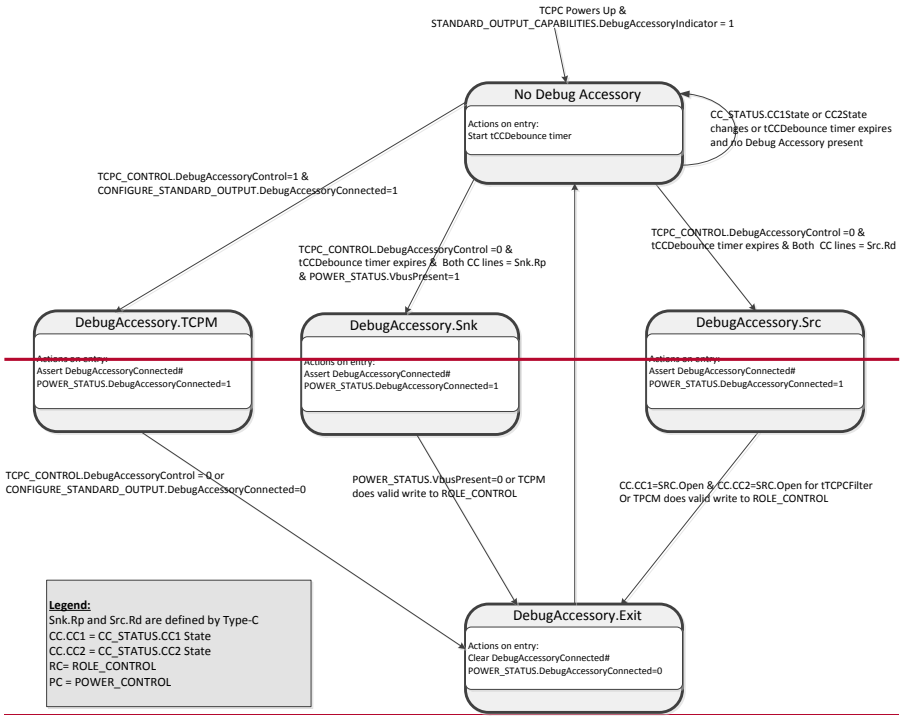
**Formatted:** Normal

**Formatted:** Keep with next

Copyright © ~~2015~~-2016 USB 3.0 Promoter Group. All rights reserved.

**Formatted:** Normal

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.




Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

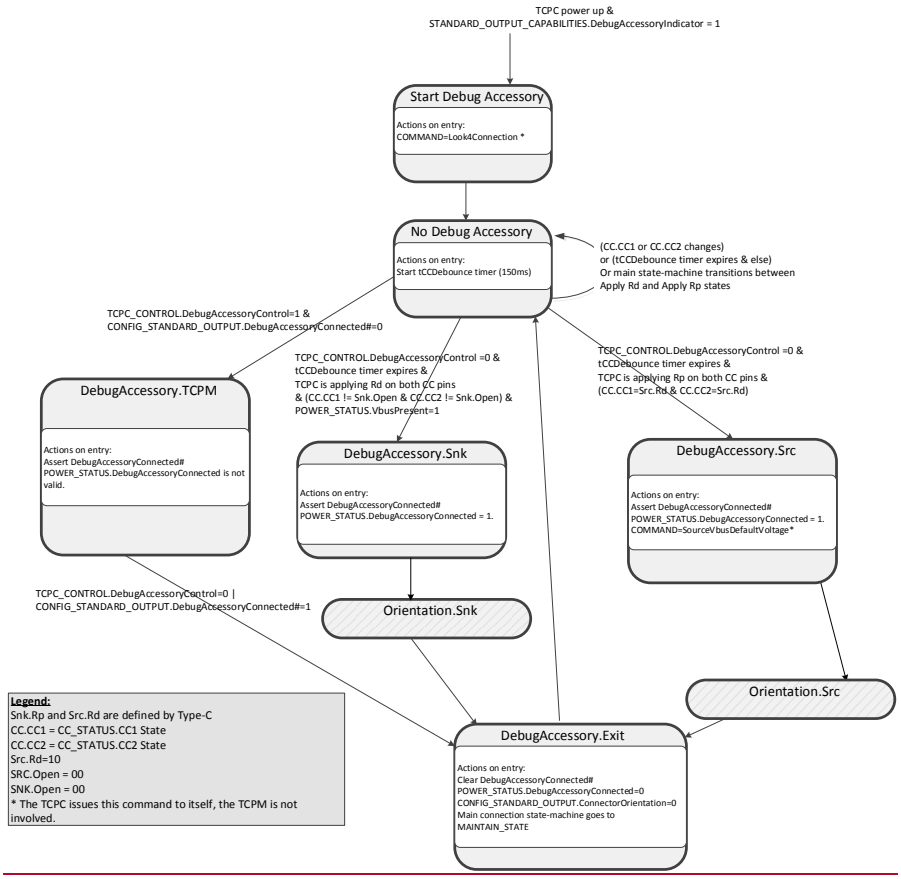



**Figure 4-~~13~~4 ~~12~~. TCPC Debug Accessory State Diagram**

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

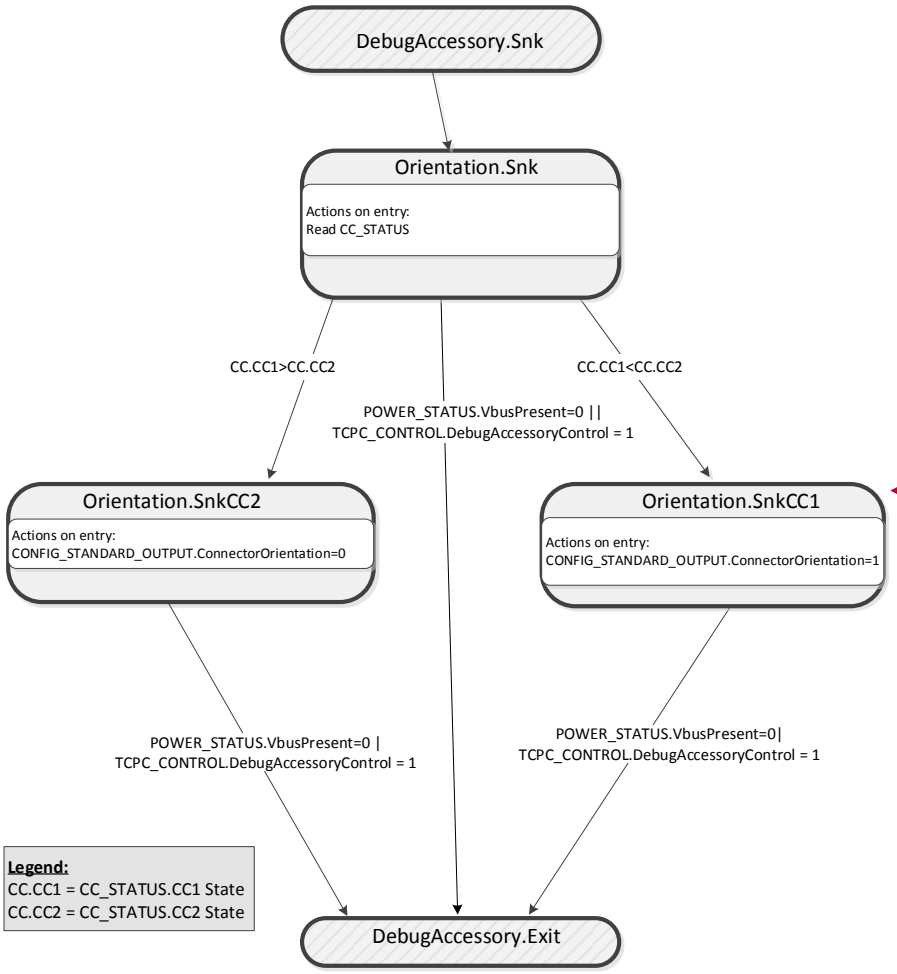



**Figure 4-14. TCPC Sink Debug Accessory Orientation State Diagram**

**Formatted:** Keep with next

**Formatted:** Left

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

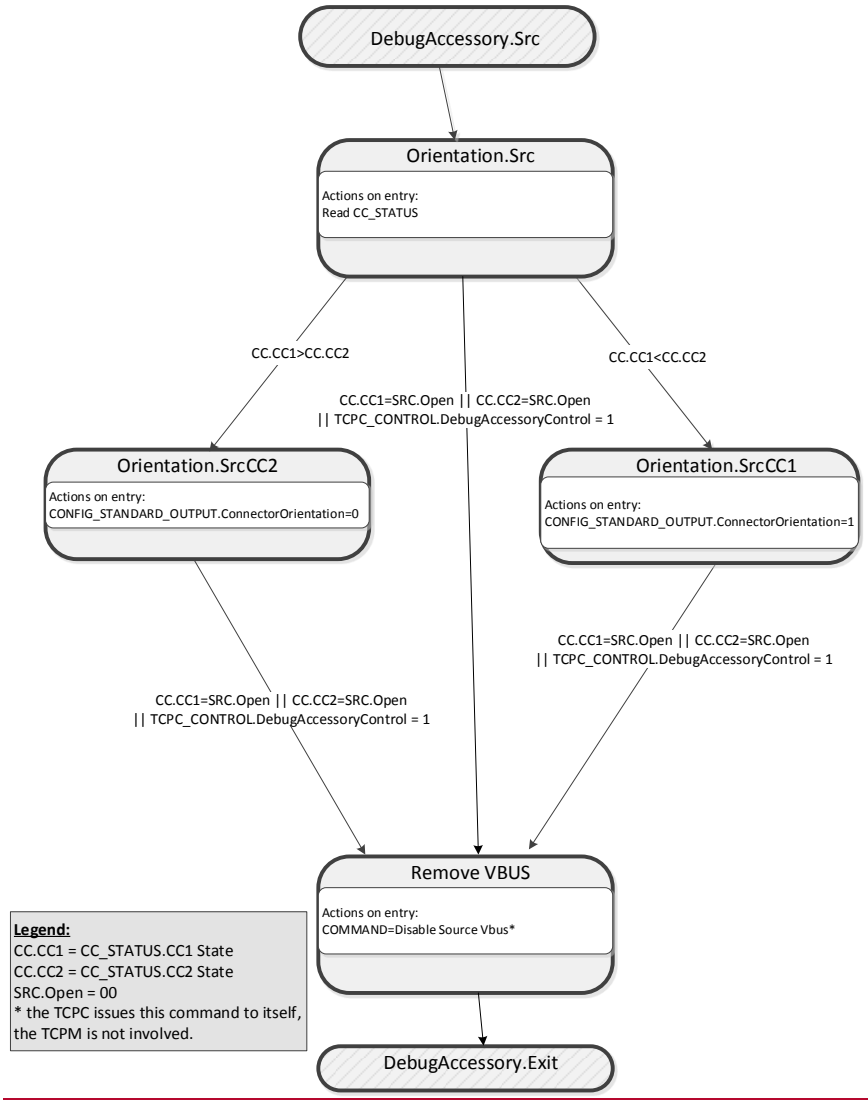



**Figure 4-15 TCPC Source Debug Accessory Orientation State Diagram**

**~~TCPC State Machine After a Connection~~**

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

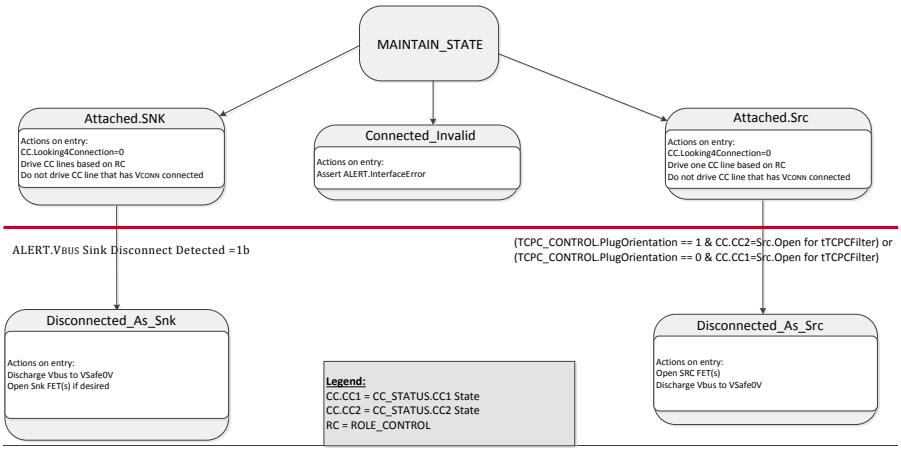
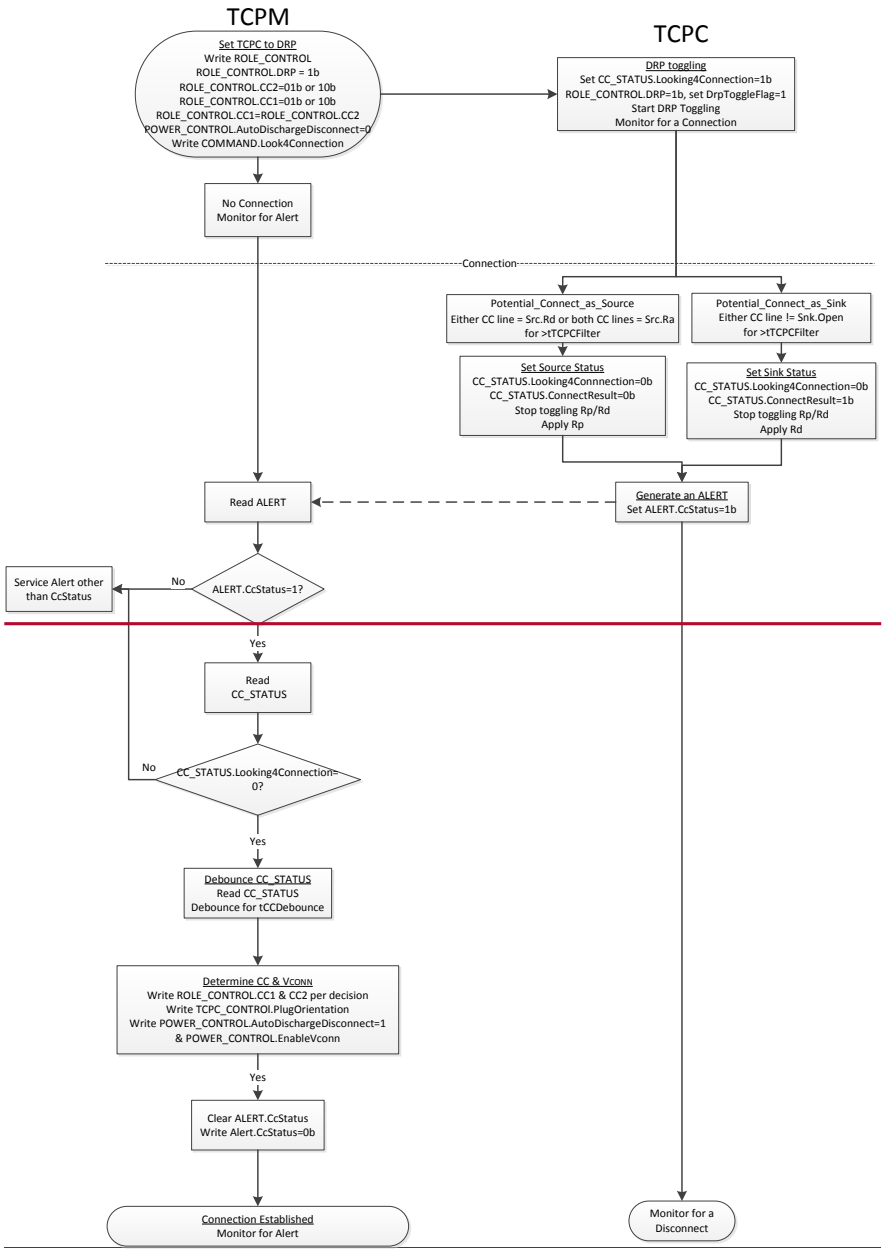



Figure 4-13. TCPC State Machine Debug Accessory

**Formatted:** Justified

Copyright © 2015-2016 USB 3.0 Promoter Group. All rights reserved.




Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

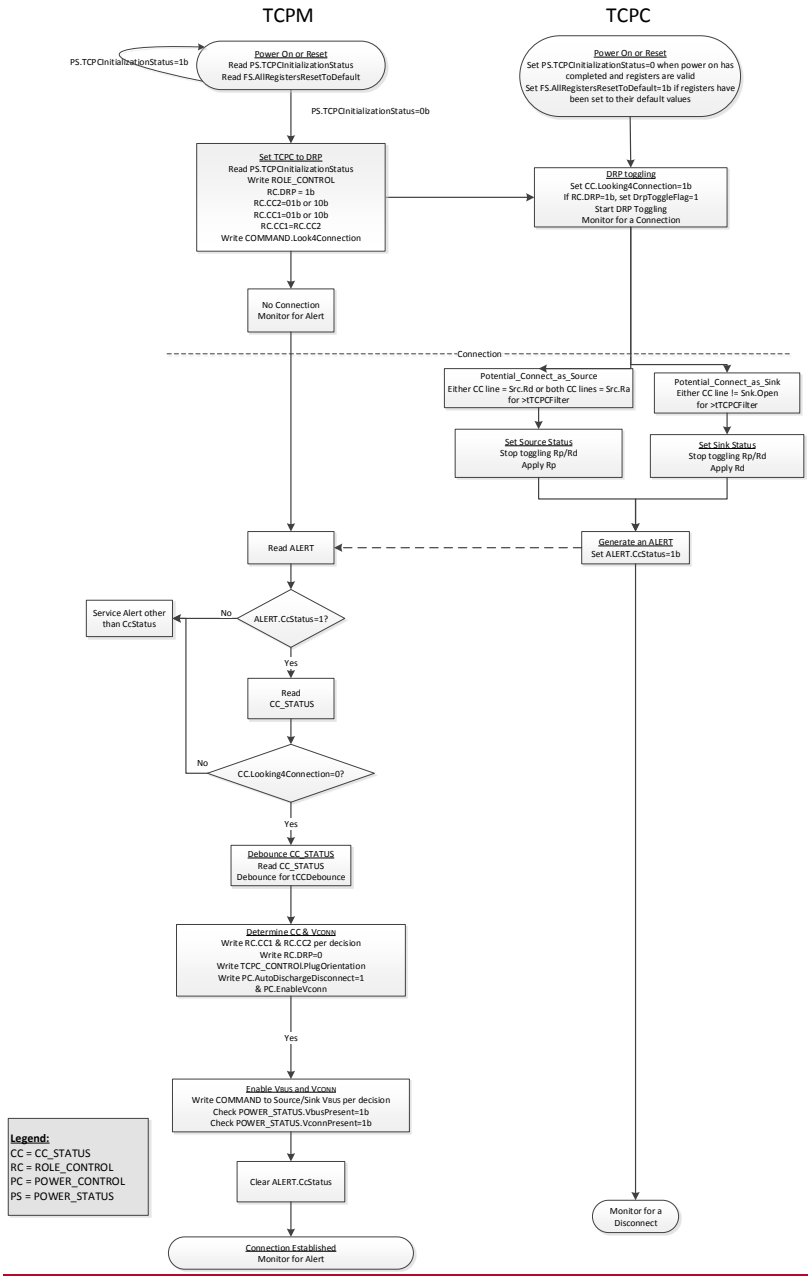
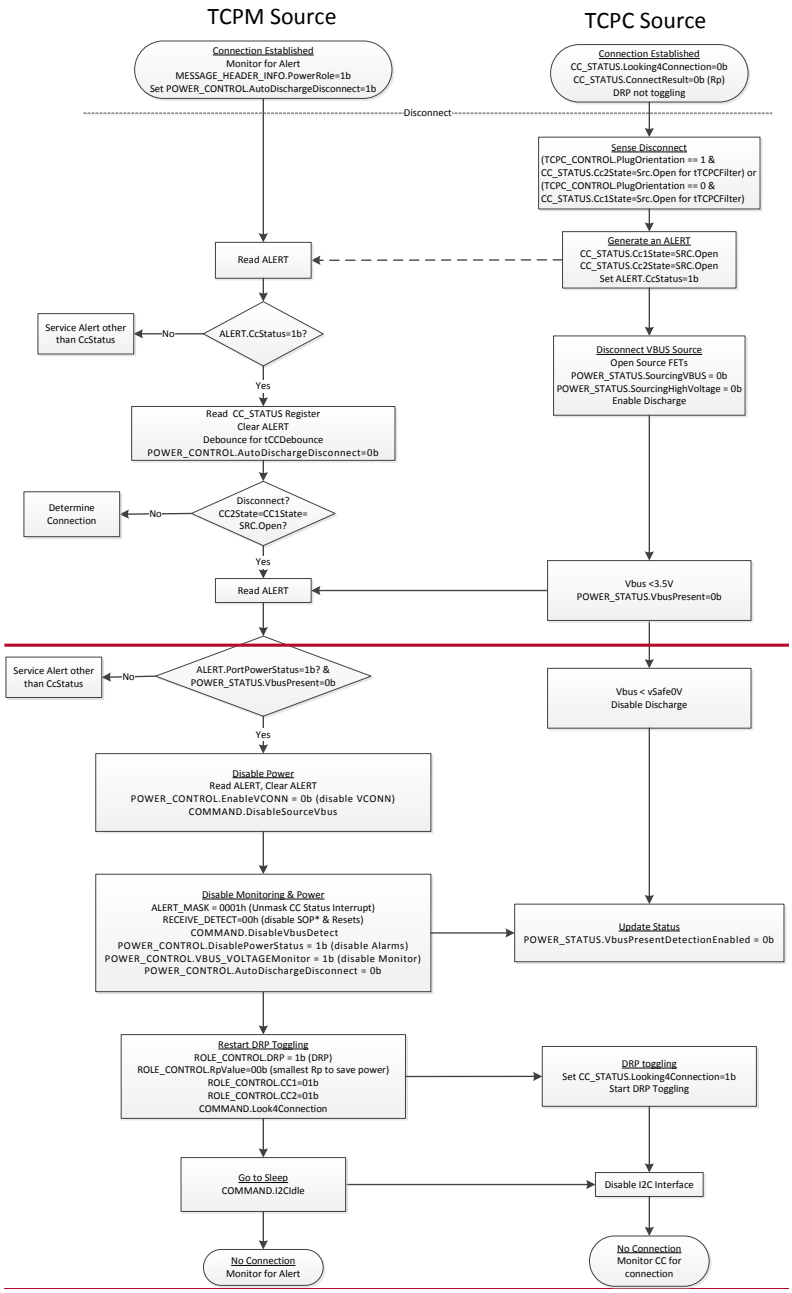



**Figure 4-~~164~~ ~~14~~. DRP Initialization and Connection Detection**

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.




Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

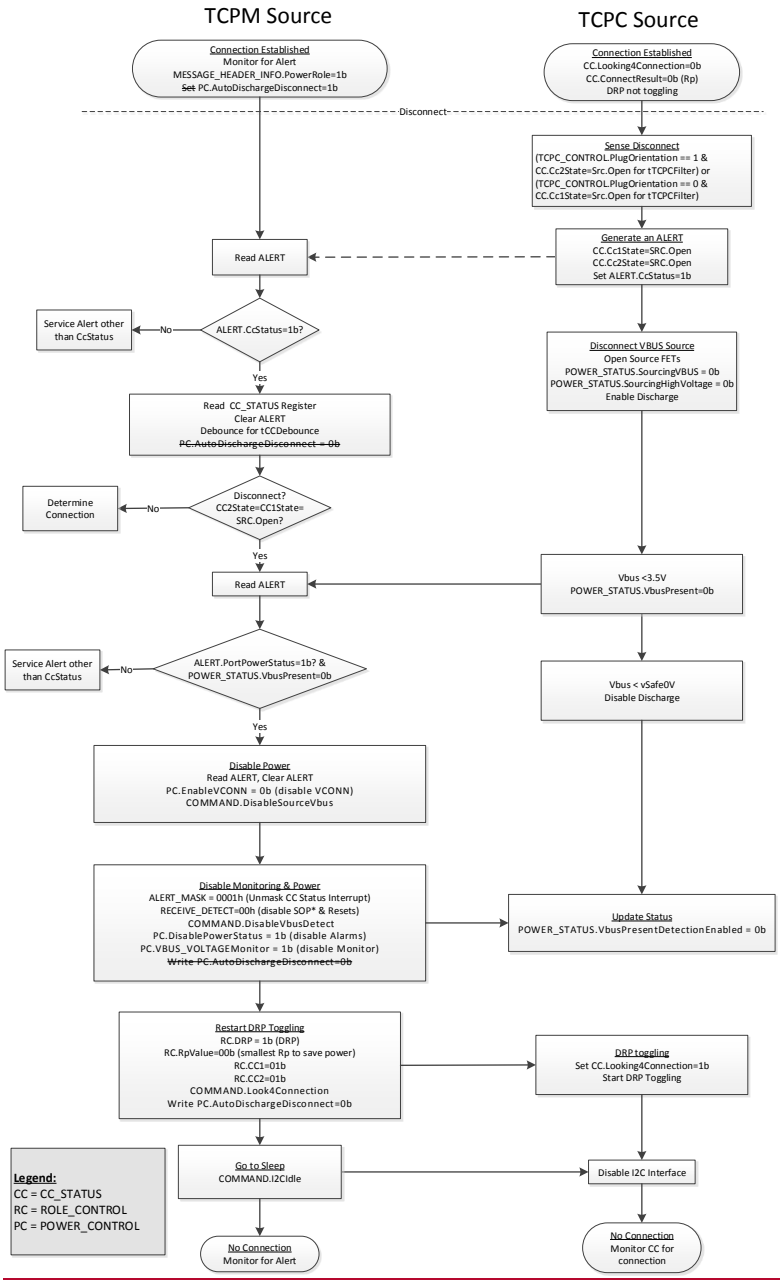
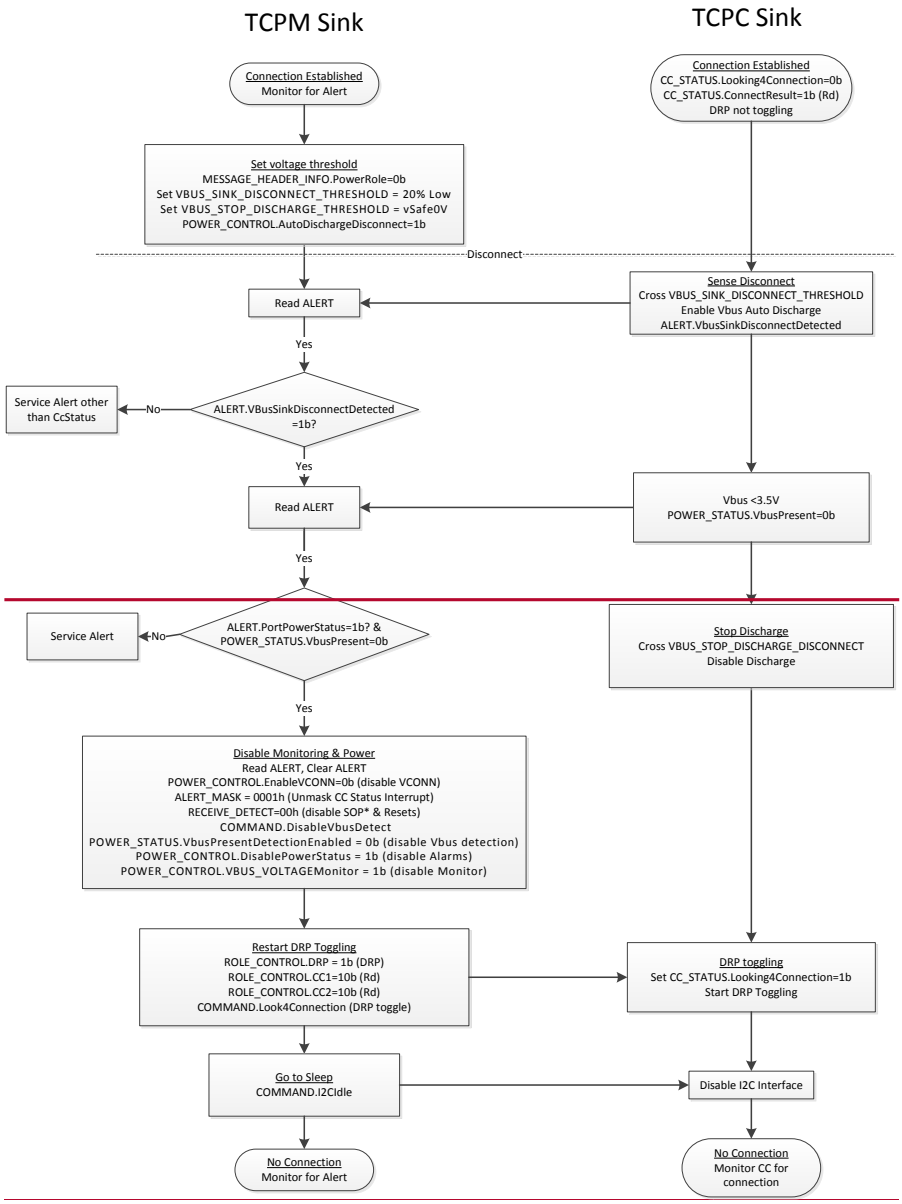



**Figure 4-~~17~~4-~~15~~. Source Disconnect**

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.




Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

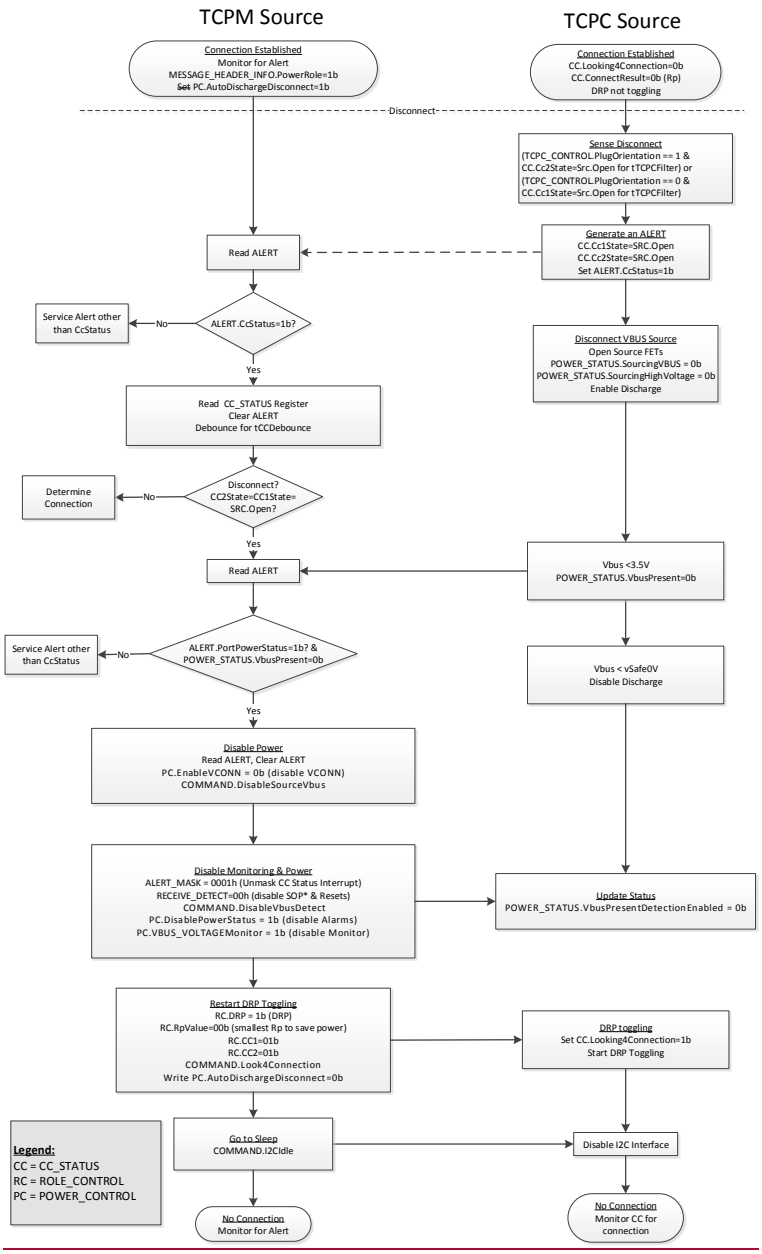



**Figure 4-~~18~~4-~~16~~. Sink Disconnect**

Copyright © ~~2015~~-2016 USB 3.0 Promoter Group. All rights reserved.

### 4.7 PD Communication Operational Model

This section describes the procedures of PD Communication through TCPC.

#### 4.7.1 Transmitting an SOP* Message

The steps for transmitting an SOP* message are as follows:

- Step 1: TCPM writes the content of the message to be transmitted into the TRANSMIT_BUFFER
- Step 2: TCPM writes to TRANSMIT requesting SOP* transmission.
- Step 3: If the TCPM writes to TRANSMIT requesting a transmission that is not Hard Reset, not Cable Reset, and not BIST Carrier Mode 2 (i.e. TRANSMIT.SOP*Message > 100b) and TRANSMIT_BYTE_COUNT is less than 3h, the TCPC shall generate a FAULT_STATUS.I2CInterfaceError.
- Step 4: The outcome of the write reported by the TCPC may be one of three indications after asserting the Alert# pin:
  - If the TCPC PHY layer successfully transmits the message, the TCPC sets the TransmitSOP*MessageSuccessful bit in the ALERT register.
  - If the TCPC PHY layer did not get any response after retries, the TCPC sets the TransmitSOP*MessageFailed bit in the ALERT register.
  - If the transmission was discarded due to an incoming message, the TCPC sets the TransmitSOP*MessageDiscarded bit in the ALERT register.
- Step 5: Before requesting another transmission, the TCPM clears the alert by writing a logical 1 to the asserted bit in the ALERT register.

When transitioning through the steps of transmitting SOP* message, the TCPC may assert ALERT.ReceiveSOP*MessageStatus or ALERT.ReceivedHardReset bit at any time to notify that a message was received.

#### 4.7.2 Transmitting a Hard Reset Message

The steps for transmitting a Hard Reset message are as follows:

- Step 1: The TCPM writes to TRANSMIT to request a Hard Reset transmission,
  - If a previous TRANSMIT request has not yet completed, the TCPC shall assert the TransmitSOP*MessageDiscarded bit in the ALERT register.
- Step 2: The TCPC asserts both ALERT.TransmitSOP*MessageSuccessful and ALERT.TransmitSOP*MessageFailed regardless of the outcome of the transmission and asserts the Alert# pin.
- Step 3: The TCPC clears the RECEIVE_DETECT and RECEIVE_BYTE_COUNT register to disable the PD message passing.
- Step 4: The TCPM clears the Alert by writing a logical 1 to the asserted bit in the ALERT register. If ALERT.ReceiveSOP*MesageStatus bit is asserted, the TCPM shall also clear the ALERT.ReceiveSOP*MessageStatus bit.
- Step 5: The TCPM writes to the RECEIVE_DETECT register to enable PD message passing.

#### 4.7.3 Receiving an SOP* message

The steps for receiving an SOP* message are as follows:

- Step1: The TCPC asserts the Alert# pin to request attention when it receives a Hard Reset, a Cable Reset, or has sent the GoodCRC in response to an SOP* message from a Port Partner. If an overflow has occurred, the TCPC will have set the ALERT.RxBufferOverflow register, therefore the TCPC will not sent the GoodCRC to other received messages until the TCPM clears the Message Received alert. The TCPM should always clear the Rx Buffer Overflow and Message Received bits at the same time. Otherwise there could be a scenario where the Rx Buffer Overflow bit remains set even though the TCPM has just cleared one of the messages in the buffer.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

- Step2: The TCPM reads the ALERT register for notification that a message was received.
- Step3: The TCPM reads the RECEIVE_BUFFER.RECEIVE_BYTE_COUNT and RECEIVE_BUFFER.RX_BUF_FRAME_TYPE. If the TCPC received an SOP* message, the TCPM reads as many bytes in the buffer (i.e. data objects in RX_BUFFER_DATA_OBJECT) as defined in the RECEIVE_BUFFER.RECEIVE_BYTE_COUNT. ~~If an overflow has occurred, the TCPC will have set the ALERT.RxBufferOverflow register.~~
- Step 4: The TCPM clears the Alerts:
  - The ALERT.RxBufferOverflow is cleared when the TCPM writes ALERT.ReceiveSOP*MessageStatus as 1 and ALERT.RxBufferOverflow as 1.
  - Writing ALERT.ReceiveSOP*MessageStatus to 1 also clears the receive buffer registers.
- Step5: After the Alert and buffers have been cleared, the TCPC shall put the next received message (if any) into the SOP* message buffer registers. The TCPC shall then update RECEIVE_BUFFER.RECEIVE_BYTE_COUNT and ALERT registers.
- Step6: If Alert# pin is still asserted, return to Step 2.

#### 4.7.4 Receiving a Hard Reset message

The TCPC steps after receiving a Hard Reset message are as follows:

- Step1: The TCPC asserts Alert# pin to request attention when it receives a Hard Reset~~, a Cable Reset, or has sent the GoodCRC in response to an SOP* message from a Port Partner.~~
- Step2: The TCPC shall set the RECEIVE_DETECT bits to zero to disable automatic transmission of GoodCRC.
- Step 3: The TCPC shall set the RECEIVE_BUFFER.RECEIVE_BYTE_COUNT to zero.

#### 4.7.5 Receiving a Cable Reset message

If the TCPC has enabled reception of Cable Reset in RECEIVE_DETECT.CableReset, the TCPC steps after receiving a Cable Reset message are as follows:

- Step1: The TCPC asserts Alert# pin to request attention when it receives a Cable Reset. The TCPC shall set Alert.ReceiveSOP*MessageStatus when it receives a Cable Reset.
- Step2: The TCPC shall set the RECEIVE_DETECT bits to zero to disable automatic transmission of GoodCRC.
- Step 3: The TCPC shall set the RECEIVE_BUFFER.RECEIVE_BYTE_COUNT to one.

**Formatted:** Space Before: 5 pt, After: 10 pt, No bullets or numbering

**Formatted:** Font: Symbol

**Formatted:** List Bullet

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### 4.8 Power Management

TCPC Interface provides the control needed to enable and disable the functional blocks in the TCPC. The purpose of disabling a functional block is to reduce the power consumption of TCPC.

Setting any bits in the ALERT_MASK register does not disable the functional blocks and has no effect on any registers.

The Alert# remains active unless all blocks are powered down and the only mechanism to wake the TCPC is via the COMMAND.I2CWake.

#### 4.8.1 I2C Interface

TCPM may put the I2C device interface in idle state by issuing COMMAND.I2CIdle. The TCPC may then generate a bit-level Not Acknowledge signal (a NAK where SDA remains HIGH during the ninth clock pulse) to its own slave address or any I2C commands.

The TCPM shall send a throw away I2C command to wake the I2C device interface in the TCPC. The TCPC shall restart its I2C interface as a side effect of seeing its slave address (even though it NAK'd). Subsequent requests sent to the slave address will not be NAK'd.

The steps for the TCPM to wake a TCPC from I2C idle:

- Step1: TCPM sends an I2C command to address the TCPC (or COMMAND.WakeI2C)
- Step2: TCPM starts tWakeI2CFail timer. If tWakeI2CFail timer expires before ALERT# is asserted, TCPM resends an I2C command to readdress the TCPC (or COMMAND.WakeI2C). The TCPC shall wake from I2C idle on the first wake command (in Step1) or the second wake command (in Step2).
- ~~Step2: The TCPC updates the I2C registers and asserts the Alert# pin~~
- Step3: The TCPC updates the ALERT.PowerStatus bit and then sets the ALERT#. The TCPC does not set any registers in the POWER_STATUS register.
- Step4: The TCPM stops the tWakeI2CFail timer.
- Step5: TCPM reads ALERT.PowerStatus registers to get notification the TCPC has exited I2C idle
- ~~Step3: TCPM reads the ALERT registers to get notification the TCPC has exited I2C idle~~
- ~~Step4: TCPM resends the last sent I2C command in Step0~~
- Step5: TCPM writes to clear ALERT registers

When the I2C device interface in the TCPC is in idle state, the TCPC shall assert the Alert# pin within ***tWakeI2CFail*** upon receiving an I2C command. The TCPM may assume an error in communication when ***tWakeI2CFail*** expires and retry issuing a throw away command to wake the TCPC.

| | Min | Max | Units |
|---|---|---|---|
| tWakeI2CFail | | 5 | ms |

**Formatted:** List Bullet, Line spacing: single, No bullets or numbering

**Formatted:** Font: (Default) Cambria

**Formatted:** Font: (Default) Cambria

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### 4.8.2 PD Messaging

The TCPM may disable PD messaging by setting the RECEIVE_DETECT and RECEIVE_BYTE_COUNT register to all zeroes. Disabling PD messaging does not cause ALERT register to be set or cleared.

### 4.8.3 CC Status Reporting

The TCPC shall disable CC Status reporting when the TCPM sets ROLE_CONTROL.DRP = 0b (No DRP) and ROLE_CONTROL.CC1 = ROLE_CONTROL.CC2 = 11b (Open). When the CC Status reporting is disabled, the TCPC shall set CC_STATUS register to all zeroes and TCPM shall ignore the values in CC_STATUS register. Disabling the CC Status reporting does not cause ALERT.CcStatus to be set or cleared.

### 4.8.4 VBUS Reporting

This section describes the operation of disabling Vbus reporting function in the TCPC.

#### 4.8.4.1 Disable VBUS Present Detection

The TCPC shall disable Vbus present detection when TCPM issues COMMAND.DisableVBUSPresentDetect. When the VBUS present detection is disabled:

- TCPC shall set POWER_STATUS.VbusPresentDetectionEnabled = 0b
- TCPM shall ignore POWER_STATUS.VBUSPresent bit
- Issuing COMMAND.DisableVBUSDetect does not cause ALERT.PowerStatus to be set or cleared

#### 4.8.4.2 Disable VBUS Voltage Alarm

The TCPC shall disable VBUS alarm reporting when the TCPM sets POWER_CONTROL.DisableVoltageAlarms = 1b. When the VBUS alarm reporting is disabled:

Setting POWER_CONTROL.DisableVoltageAlarms = 1b has no effect on POWER_STATUS register.

#### 4.8.4.3 Disable VBUS Monitoring

The TCPC shall disable VBUS monitor reporting when the TCPM sets POWER_CONTROL.VBUS_VOLTAGEMonitor = 1b. When the VBUS_VOLTAGE monitor reporting is disabled:

- The TCPC shall set VBUS_VOLTAGE register to all zeroes and TCPM shall ignore the values in VBUS_VOLTAGE register
- Setting POWER_CONTROL.VBUS_VOLTAGEMonitor = 1b has no effect on POWER_STATUS register

#### 4.8.4.4 Disable VBUS Auto Discharge

For a Source, Auto Discharge is enabled/disabled by writing to POWER_CONTROL,AutoDischargeDisconnect.

For a Sink, Auto Discharge is enabled by setting non zero to VBUS_SINK_DISCONNECT_THRESHOLD and set POWER_CONTROL,AutoDischargeDisconnect to one.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

For a Sink, Auto Discharge is disabled by either setting all zeros to VBUS_SINK_DISCONNECT_THRESHOLD zero.

### 4.8.5 Fault Status Reporting

The TCPM may disable individual FAULT_STATUS reporting by setting the appropriate FAULT_CONTROL bit to 1. The TCPM may indicate to the TCPC to power down all the FAULT circuitry by setting FAULT_CONTROL to all 1s.

When the fault status reporting is disabled:

- When a bit in FAULT_CONTROL is set to one, the TCPC shall set the corresponding bit in the FAULT_STATUS register to zero and TCPM shall ignore that bit.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

### 4.9 TCPC Timing Constraints

The TCPC shall comply with the timing constraints defined in Table 4-45.

In a scenario where all ports receive a message with 7 data objects simultaneously, and each port is required to respond with a message containing 7 data objects within tReceiverResponse on each port, the actual value of the timing parameters below affects the number of ports that can be supported. If timing constraints are met, the TCPM may support up to 4 ports with the following assumptions:

- The host operates the I2C at $F_{SCL}$ = 1MHz
- The host requires 5ms or less to enter TCPM task and start servicing PD Messages of the 4 ports

The host requires 1ms or less to respond to a PD Message while servicing upper layer interrupts and maintaining states.

**Table 4-~~45~~4-~~45~~. TCPC Timing Constraints**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| tBUFFER2CC | Time between I2C STOP and first bit of SOP | | 195 | us |
| tCc2BUFFER | Time between last bit of EOP and Rx buffer ready | | 50 | us |
| tSetReg | Time between status change occurs and status register(s) updated | | 50 | us |
| tCcStatusDelay | Time between status change occurs and the CC wire stabilizes | | 200 | us |
| tHVWatchdog | Time from last I2C transaction or ALERT# pin assertion to entering ErrorRecovery | 650 | 5000 | ms |

### 4.10 I2C Physical Interface Specifications

The I2C interface shall follow the electrical specifications defined in "I2C-bus specification and user manual Rev.6" (4th April 2014) **http://www.nxp.com/documents/user_manual/UM10204.pdf** except for the parameters defined in Table 4-46 and Table 4-47.

The I2C interface shall be compatible with Fast-mode Plus and is defined at a bit rate at 1Mbit/s. The TPCM may choose to run at a slower rate that Fast-mode Plus, but the TCPC must be capable of operating at the Fast-mode Plus rate. The TCPC is allowed limited clock stretching by holding the SCL line low. The TCPC shall not increase the duration of a single-byte read by more than tI2C_SBR. The TCPC shall not increase the duration of a single-byte write by more than tI2C_SBW. The TCPC shall not increase the duration of a multi-byte read by more than tI2C_MBR. The TCPC shall not increase the duration of a multi-byte write by more than tI2C_MBW. The TCPC enable or disable clock stretching as defined in Section 4.4.5.1.

Each byte on I2C is really 9 bits – 8 data bits followed by ACK/NAK. Write transfers have 2 bits of additional overhead for the start [S]/stop[P] bits. Read transfers also have an additional Repeated Start [Sr] overhead bit (3 total overhead bits), plus they send the i2c address byte twice (once to write the register offset, the 2nd time to send the read command)

The I2C interface shall be implemented with hysteresis and a Schmitt trigger. I2C IOs are open-drain. The I2C interface specifications are defined over a voltage range of 1.7V to 3.3V.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

**Table 4-~~46~~4-~~46~~. I2C Static Characteristics**

| Symbol | Parameter | Conditions | Min | Max | Units |
|---|---|---|---|---|---|
| $V_{DD}$ | I/O Supply Voltage | | 1.8 | 3.6 | Volts |
| $V_{IH}$ | HIGH-level input voltage | | 0.7$V_{DD}$ | $V_{DD}$+0.5 | Volts |
| $I_I$ | Input current each IO Pin | 0.1$V_{DD}$ < $V_{IL}$ or $V_{IH}$ < 0.9$V_{DDMAX}$ | -10[3] | +10[3] | uA |

Note:

1. At 3mA sink current, VDD > 2V
2. At 2mA sink current, VDD<= 2V
3. If $V_{DD}$ is switched off, IO pins shall not obstruct the SDA and SCL Lines.

**Table 4-~~47~~4-~~47~~. I2C Dynamic Characteristics**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $F_{SCL}$ | SCL Clock Frequency | 400 | 1000 | KHz |
| $t_{SP}$ | Pulse width of spikes that must be suppressed by the input filter | | 50[2] | ns |
| $t_{LOW}$ | LOW period of the SCL clock | 0.5 | - | us |
| $t_{HIGH}$ | HIGH period of the SCL clock | 0.26 | - | us |
| $t_{HD:DAT}$ | Data hold time | 0 | - | us |
| $t_{SU:DAT}$ | Data set-up time | 50 | - | ns |
| $t_R$ | Rise time of both SDA and SCL signals | - | 120 | ns |
| $t_F$ | Fall time of both SDA and SCL signals | 20x ($V_{DD}$/5.5)[1] | 120 | ns |
| $t_{BUF}$ | Bus free time between a STOP and START condition | 0.5 | - | us |
| $C_B$ | Capacitance load for each bus line[3] | - | 550 | pF |
| $t_{VD:DAT}$[4] | Data valid time | - | 0.45 | us |
| $t_{VD:ACK}$[5] | Data valid acknowledge time | - | 0.45 | us |
| $t_{I2C_SBR}$ (1000KHz) | Time for I2C SINGLE BYTE READ | - | 50 | us |
| $t_{I2C_SBW}$ (1000KHz) | Time for I2C SINGLE BYTE WRITE | - | 40 | us |
| $T_{I2C_MBR}$ (1000KHz) | Time for I2C Multi BYTE READ | - | 50 + 12/byte[7] | us |
| $T_{I2C_MBW}$ (1000KHz) | Time for I2C Multi BYTE WRITE | | 40 + 12/byte[7] | us |
| $t_{I2C_SBR}$ (400KHz)[6] | Time for I2C SINGLE BYTE READ | - | 110 | us |
| $t_{I2C_SBW}$ (400KHz)[6] | Time for I2C SINGLE BYTE WRITE | - | 85 | us |
| $T_{I2C_MBR}$ (400KHz)[6] | Time for I2C Multi BYTE READ | - | 100 + 35/byte[7] | us |
| $T_{I2C_MBW}$ (400KHz)[6] | Time for I2C Multi BYTE WRITE | | 85 + 30/byte[7] | us |

1. Necessary to be backwards compatible with Fast-mode.
2. Input filters on the SDA and SCL inputs suppress noise spikes of less than 50ns.
3. The maximum capacitance allowable may vary from this value depending on the actual operating voltage and frequency of the application.
4. Time for data signal from SCL LOW to SDA output (HIGH or LOW, depending on which one is worse).

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.

5. Time for acknowledgement signal from SCL LOW to SDA output (HIGH or LOW, depending on which one is worse).
6. The TCPC should only run at 400KHz Fscl when the TCPM is servicing only one TCPC.
7. The TCPM may disable clock stretching by setting TCPC_CONTROL.I2CClockStretchingControl to 00b. The TCPC is not allowed to Nak I2C transfers no matter which clock stretching setting is chosen by the TCPM, unless the TCPM has put it to sleep using COMMAND.I2CIdle.

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

## A Informative TCPM State~~ Machine~~ Diagrams~~ Diagrams~~

This section provides informative TCPM Protocol Layer ~~S~~state ~~Machine Diagrams~~ diagrams to aid the TCPM designer. The ~~State Machines~~state diagrams are not intended to indicate requirements, but rather to give guidance on the interaction between TCPC and TCPM. The TCPM should follow ~~*USB PDUSB PD*~~ if receiving an unexpected GoodCRC.

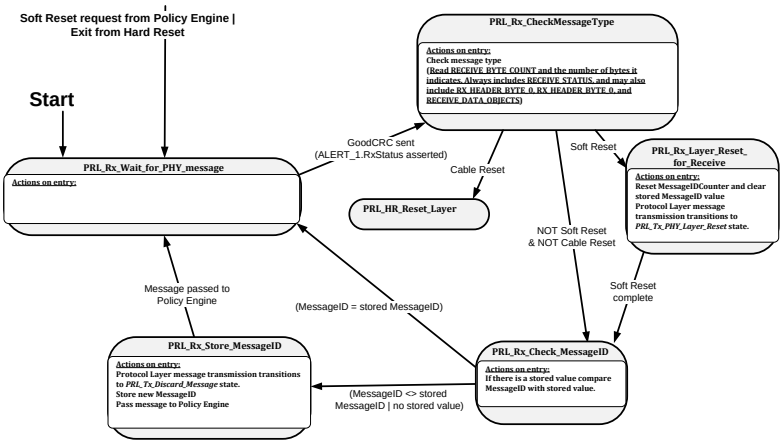



**Figure 4-19. Rx State Diagram Implemented in TCPM**

~~**Figure A.1. Rx State Machine Implemented in TCPM**~~

**Formatted:** bookmark Char, Font: Not Bold

**Formatted:** Check spelling and grammar

**Formatted:** Check spelling and grammar

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

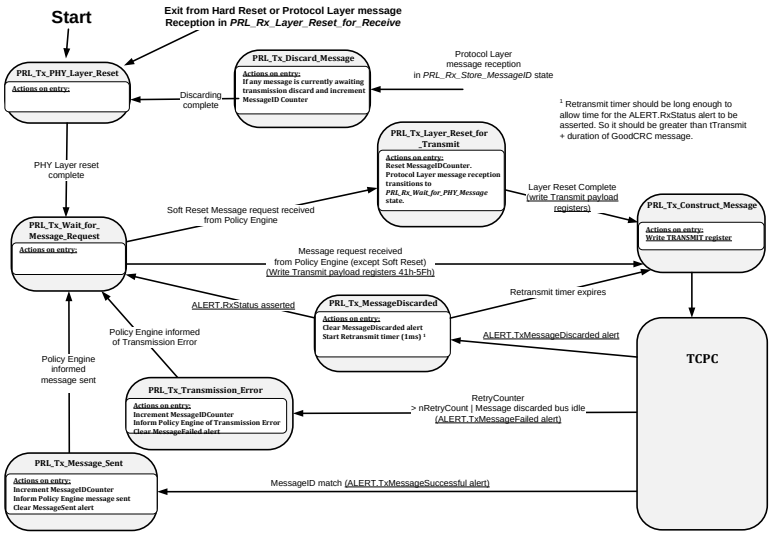
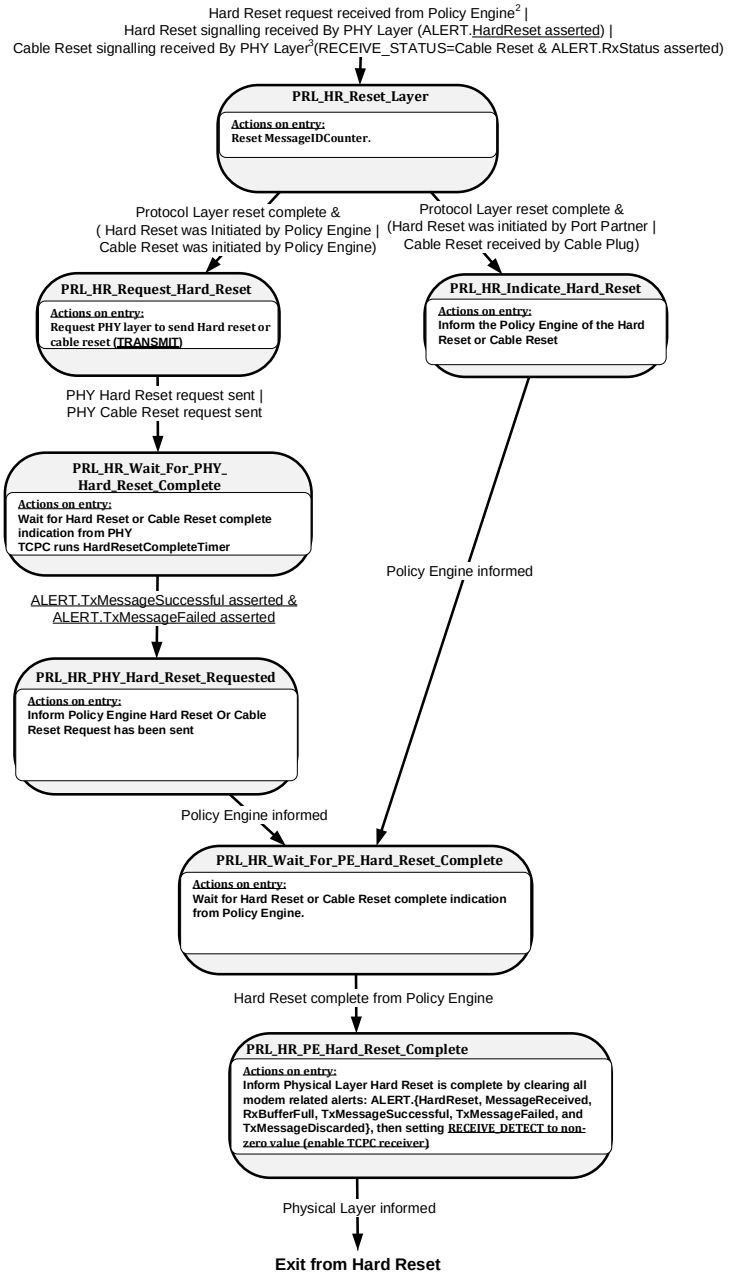



**Figure 4-20. Tx State Diagram Implemented in TCPM**

~~**Figure A-2. Tx State Machine Implemented in TCPM**~~

Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.




Copyright © ~~2015~~ 2016 USB 3.0 Promoter Group. All rights reserved.

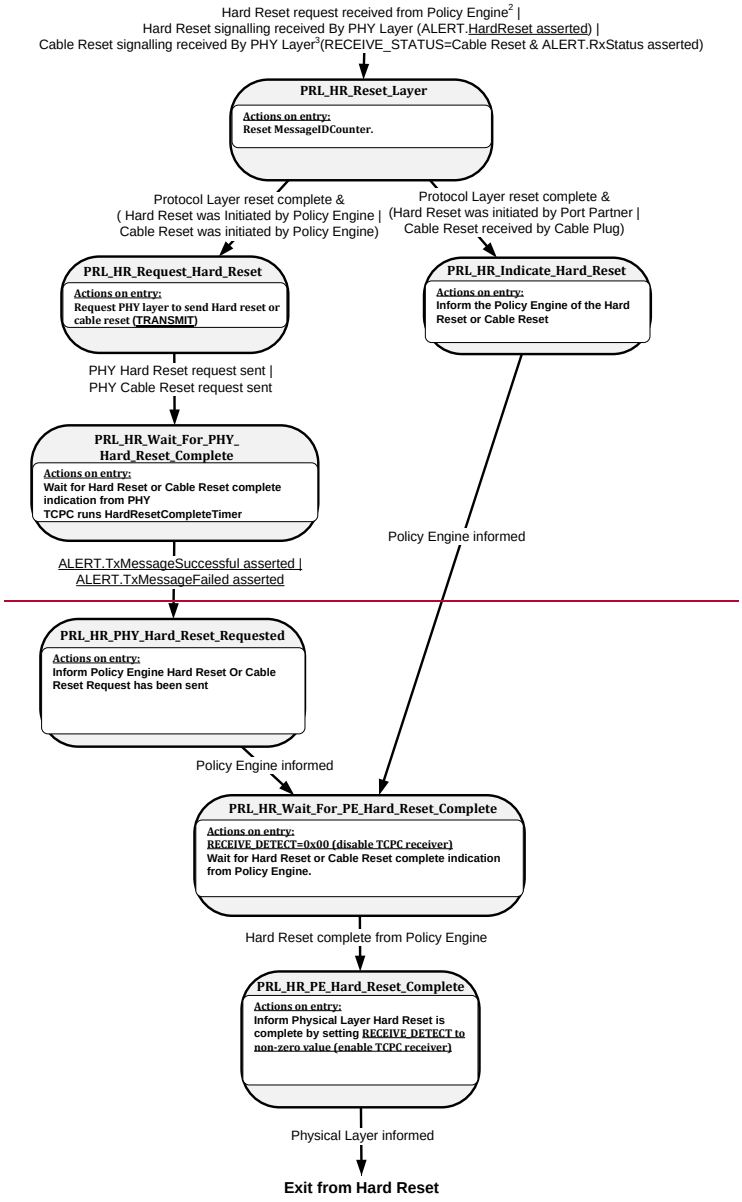



**Figure 4-21. Hard Reset State Diagram Implemented in TCPM** ~~**Figure A-3. Hard Reset State Machine Implemented in TCPM**~~

**Formatted:** Caption

Copyright © ~~2015~~2016 USB 3.0 Promoter Group. All rights reserved.