Universal Serial Bus Type-C® Port Controller Interface Specification

Revision 2.0, Version 1.3
January 2022

CONTENTS

Specification Work Group Chairs / Specification Editors 9
Specification Work Group Contributors 9
Revision, Version History 11
1 Introduction 12
1.1 Purpose 12
1.2 Scope 12
1.3 Related Documents 13
1.4 Conventions 13
1.4.1 Precedence 13
1.4.2 Keywords 13
1.4.3 Numbering 14
1.5 Terms and Abbreviations 14
2 Overview 15
2.1 Introduction 15
2.2 USB Type-C Port Controller (TCPC) Interface 15
2.3 Changes from Revision 1.0 15
2.4 Changes from Revision 2.0 Version 1.0 16
2.5 Changes from Revision 2.0 Version 1.1 16
2.6 Changes from Revision 2.0 Version 1.2 16
3 USB Type-C® Port Controller Requirements 18
3.1 Port Power Control for VBUS and VCONN 18
3.2 USB CC Logic 18
3.3 USB-PD Message Delivery 18
3.4 Debug Accessory Detection (Optional Normative) 19
3.5 State Diagram Introduction 20
3.6 TCPC Protocol Layer State Operation 21
3.7 Source, Sink, and DRP Requirements 23
3.7.1 Source Requirements 27
3.7.2 Sink Requirements: 28
3.7.3 Sink with Accessory Support 29
3.7.4 DRP Requirements 30
3.8 Watchdog Timer Requirements (Optional Normative) 31
3.8.1 Watchdog Timer Function 31
4 USB Type-C® Port Controller Interface 32
4.1 SMBus with Packet Error Checking Mechanism (Optional Normative) 32
4.2 Register Map 33
4.3 Writing and Reading Registers 36
4.3.1 Writing Single Byte Registers 36
4.3.2 Reading Single Byte Registers 37
4.3.3 Writing Two-Byte Registers 37
4.3.4 Reading Two-Byte Registers 38

4.3.5 Writing the TRANSMIT_BUFFER 38
4.3.6 Reading the RECEIVE_BUFFER 38
4.3.7 Writing Single Byte Registers using SMBus with PEC 39
4.3.8 Reading Single Byte Registers using SMBus with PEC 40
4.3.9 Writing Two-Byte Registers using SMBus with PEC 40
4.3.10 Reading Two-Byte Registers using SMBus with PEC 41
4.3.11 Writing the TRANSMIT_BUFFER using SMBus with PEC 41
4.3.12 Reading the RECEIVE_BUFFER using SMBus with PEC 41
4.4 Register Definition 42
4.4.1 Identification Registers 42
4.4.2 ALERT Register (Normative) 44
4.4.3 Mask Registers 46
4.4.4 CONFIGURE STANDARD OUTPUT (Optional Normative) 50
4.4.5 Control and Configuration Registers 51
4.4.6 Status Registers 61
4.4.7 ALERT_EXTENDED (Normative) 65
4.4.8 COMMAND (Normative) 67
4.4.9 Capability Registers 71
4.4.10 CONFIGURE EXTENDED1 (Optional Normative) 76
4.4.11 GENERIC_TIMER (Optional Normative) 76
4.4.12 MESSAGE_HEADER_INFO (Normative) 78
4.4.13 RECEIVE_DETECT (Normative) 78
4.4.14 RECEIVE_BUFFER (Normative) 79
4.4.15 TRANSMIT (Normative) 80
4.4.16 TRANSMIT_BUFFER (Normative) 81
4.4.17 VBUS_VOLTAGE (Optional Normative) 82
4.4.18 Voltage Thresholds 83
4.4.19 VBUS_NONDEFAULT_TARGET (Optional Normative) 85
4.4.20 VENDOR_DEFINED Registers 85
4.5 STANDARD IO SIGNALS 86
4.5.1 STANDARD INPUT SIGNALS (Optional Normative) 86
4.5.2 STANDARD OUTPUT SIGNALS (Optional Normative except Alert#) 86
4.6 Type-C Port Controller Connection State Diagrams and Flows 88
4.7 USB PD Communication Operational Model 97
4.7.1 Transmitting an SOP* USB PD Message with Less than or Equal to 128 Data Bytes 97
4.7.2 Transmitting an SOP* USB PD Message with Greater than 128 Data Bytes 97
4.7.3 Transmitting a Hard Reset Message 100
4.7.4 Transmitting a Cable Reset Message 100
4.7.5 Receiving SOP* USB PD Messages with Less than or Equal to 128 Data Bytes 100
4.7.6 Receiving SOP* USB PD Messages with Greater than 128 Data Bytes 101

4.7.7 Re-Reading RECEIVE_BUFFER....102
4.7.8 Receiving a Hard Reset message....104
4.7.9 Receiving a Cable Reset message....104
4.8 Power Management....105
4.8.1 I2C Interface....105
4.8.2 USB PD Message Delivery....105
4.8.3 CC Status Reporting....105
4.8.4 VBUS Reporting....106
4.8.5 Fault Status Reporting....106
4.9 Type-C Port Controller Timing Constraints....107
4.10 I2C Physical Interface Specifications....107
A Informative TCPM Protocol Layer State Diagrams....109
B Informative TCPM Timing Considerations....111
C TCPM Timing Constraints when Acting as a Source and Setting SinkTxOK....113

FIGURES

Figure 1-1. USB Type-C Port Manager to USB Type-C Port Controller Interface 12
Figure 2-1. TCPC Interface 15
Figure 3-1. Outline of States 20
Figure 3-2. Reference to States 20
Figure 3-3. Message Reception State Diagram implemented in TCPC 21
Figure 3-4. Message Transmission State Diagram Implemented in TCPC 22
Figure 3-5. Hard Reset Transmission State Diagram Implemented in TCPC 22
Figure 4-1. Writing Consecutive Single Byte Registers with or without the SMBUS Protocol 36
Figure 4-2. Reading Consecutive Single Byte Registers with or without the SMBus Protocol 37
Figure 4-3. Writing a 2-Byte Register with or without the SMBus Protocol 37
Figure 4-4. Reading a 2-Byte Register with or without the SMBus Protocol 38
Figure 4-5. Writing the TRANSMIT_BUFFER with or without the SMBus Protocol 38
Figure 4-6. Reading the RECEIVE_BUFFER with or without the SMBus Protocol 38
Figure 4-7. Writing Consecutive Single Byte Registers using SMBus PEC 39
Figure 4-8. Reading Consecutive Single Byte Registers using SMBus PEC 40
Figure 4-9. Writing a 2-Byte Register using SMBus PEC 40
Figure 4-10. Reading a 2-Byte Register using SMBus PEC 41
Figure 4-11. Writing the TRANSMIT_BUFFER using SMBus PEC 41
Figure 4-12. Reading the RECEIVE_BUFFER using SMBus PEC 41
Figure 4-13. Automatic VBUS Sink Discharge by the TCPC after a Disconnect 57
Figure 4-14. COMMAND.SendFRSwapSignal triggered Fast Role Swap operation 59
Figure 4-15. STANDARD INPUT SIGNAL Source FR Swap triggered Fast Role Swap operation 60
Figure 4-16. Transition from vSafe5V to Nondefault Voltage 69
Figure 4-17. Transition from Nondefault Voltage to vSafe5V 70
Figure 4-18. TCPC Power-On State Diagram 88
Figure 4-19. TCPC Main State-Machine: Before a Connection 89
Figure 4-20. TCPC Main State-Machine: After a Connection 90
Figure 4-21. TCPC Debug Accessory State-Machine 91
Figure 4-22. TCPC Debug Accessory State-Machine: Sink Orientation 92
Figure 4-23 TCPC Debug Accessory State-Machine: Source Orientation 93
Figure 4-24. DRP Initialization and Connection Detection 94
Figure 4-25. Source Disconnect 95
Figure 4-26. Sink Disconnect 96
Figure A-1. Message Reception State Diagram Implemented in TCPM 109
Figure A-2. Message Transmission State Diagram Implemented in TCPM 109
Figure A-3. Hard Reset State Diagram Implemented in TCPM 110

TABLES

Table 3-1. Required DEVICE_CAPABILITIES_1 Support 24
Table 3-2. Required DEVICE_CAPABILITIES_2 Support 25
Table 3-3. Source Requirements 27
Table 3-4. Sink Requirements 28
Table 3-5. Sink with Accessory Support Requirements 29
Table 3-6. DRP Requirements 30
Table 4-1. Register Map 33
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 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. EXTENDED_STATUS_MASK Register Definition 48
Table 4-13. ALERT_EXTENDED_MASK Register Definition 48
Table 4-14. CONFIG_STANDARD_OUTPUT Register Definition 50
Table 4-15. TCPC_CONTROL Register Definition 51
Table 4-16. ROLE_CONTROL Register Definition 53
Table 4-17. Power On Default Conditions 53
Table 4-18. FAULT_CONTROL Register Definition 54
Table 4-19. POWER_CONTROL Register Definition 55
Table 4-20. Discharge Timing Parameters 56
Table 4-21. Debounce requirements 61
Table 4-22. CC_STATUS Register Definition 62
Table 4-23. POWER_STATUS Register Definition 63
Table 4-24. FAULT_STATUS Register Definition 64
Table 4-25. EXTENDED_STATUS Register Definition 65
Table 4-26. ALERT_EXTENDED Register Description 66
Table 4-27. COMMAND Register Definition 68
Table 4-28. DEVICE_CAPABILITIES_1 Register Definition 71
Table 4-29. DEVICE_CAPABILITIES_2 Register Definition 72
Table 4-30. DEVICE_CAPABILITIES_3 Register Definition 74
Table 4-31. STANDARD_INPUT_CAPABILITIES Register Definition 75
Table 4-32. STANDARD_OUTPUT_CAPABILITIES Register Definition 75
Table 4-33. CONFIG_EXTENDED1 Register Definition 76
Table 4-34. GENERIC_TIMER Register Definition 77
Table 4-35. MESSAGE_HEADER_INFO Register Definition 78
Table 4-36. RECEIVE_DETECT Register Definition 79
Table 4-37. READABLE_BYTE_COUNT Definition 80
Table 4-38. RX_BUF_FRAME_TYPE Definition 80
Table 4-39. TRANSMIT Register Definition 81
Table 4-40. I2C_WRITE_BYTE_COUNT Definition 82
Table 4-41. VBUS_VOLTAGE Register Definition 82
Table 4-42. VBUS_SINK_DISCONNECT_THRESHOLD Register Description 83
Table 4-43. VBUS_STOP_DISCHARGE_THRESHOLD Register Description 84
Table 4-44. VBUS_VOLTAGE_ALARM_HI_CFG Register Description 84
Table 4-45. VBUS_VOLTAGE_ALARM_LO_CFG Register Description 84
Table 4-46. VBUS_NONDEFAULT_TARGET Register Description 85
Table 4-47. STANDARD INPUT SIGNALs 86

Table 4-48. STANDARD OUTPUT SIGNALs 86
Table 4-49. TCPC Timing Constraints 107
Table 4-50. I2C Static Characteristics 107
Table 4-51. I2C Dynamic Characteristics 108
Table 4-52. Implementations and Impact on TCPCs on one I2C Interface 111
Table 4-53. TCPC Alert Servicing Timing 113

Specification Work Group Chairs / Specification Editors

Intel Corporation (USB 3.0 Promoter company)	Dmitriy Berchanskiy – Subgroup WG co-chair, Specification Co-author
Specwerkz (USB 3.0 Promoter company)	Robert Dunstan – Subgroup WG co-chair, Specification Co-author

Specification Work Group Contributors

Advanced Micro Devices	Will Harris Jason Hawken	Joseph Scanlon Peter Teng	Ken Xue
Analogix Semiconductor, Inc.	Greg Stewart		
Apple	Sree Anantharaman William Ferry Reese Schreiber	Colin Whitby-Strevens Jeff Wilcox Dan Wilson	Hsiao-Ping Jennifer Tsai Scott Jackson
Cadence Design Systems, Inc.	Jacek Duda Pawel Eichler	Michal Staworko	
Canova Tech	Matteo Casalin	Davide Ghedin	Nicola Scantamburlo
Cypress Semiconductor	Anup Nayak Jagadeesan Raj	Rushil Kadakia Ganesh Subramaniam	
Dell Inc.	Mohammed Hijazi Marcin Nowak	Siddhartha Reddy Merle Wood	
Diodes Inc	Kay Annamalai		
DisplayLink (UK) Ltd.	Pete Burgers Dan Ellis	Kevin Jacobs Richard Petrie	Christo Welgemoed
Ellisys	Mario Pasquali	Chuck Trefts	
Etron Technology, Inc.	Shihmin Hsu	Chien-Cheng Kuo	
Fresco Logic Inc.	Tim Barilovits Bob McVay	Dean Susnow Jeffrey Yang	Christopher Meyers
Google Inc.	Alec Berg Abe Levkoy Jim Guerin Mark Hayter	Sameer Nanda Vincent Palatin David Schneider	Scott Collyer
Granite River Labs	Mike Engbretson		
Hewlett Packard Inc	Roger Benson	Robin Castell	
Huawei Technologies Co., Ltd.	Chao Xue		
Indie Semiconductor	Robert Heaton		
Intel Corporation	Dmitriy Berchanskiy Basavaraj Astekar Jenn Chuan Cheng Tim McKee	Abdul Ismail Brad Saunders Henrik Leegaard	Vijaykumar Kadgi Sanjeev Jahagirdar Karthi Vadivelu
Keysight Technologies Inc.	Jit Lim		
Lattice Semiconductor Corp	Young Il Kim		

Maxim Integrated Products	Jason Ferguson	Jacob Scott	
MCCI Corporation	Terry Moore	Sherri Russo	Yogender Saroha
Microchip Technology Inc.	Josh Averyt	Mark Bohm	Shannon Cash
	John Sisto	Richard Wahler	Richard Petrie
Microsoft Corporation	Randy Aull	Michelle Bergeron	Anthony Chen
	Vivek Gupta	David Hargrove	Robbie Harris
	Teemu Helenius	Kai Inha	Jayson Kastens
	Ismo Manninen	Rahul Ramadas	Nathan Sherman
	Tatu Tomppo	Andrew Yang	
MQP Electronics Ltd.	Sten Carlsen	Pat Crowe	
Nuvoton Technology Corp	Victor Flachs	Tomer Levi	Amit Maoz
	Nimrod Peled		
NXP Semiconductors	Ken Jaramillo	Anand Kannan	Vijendra Kuroodi
	Abhijeet Kulkarni	Krishnan TN	Bart Vertenten
	Rod Whitby	Dennis Ha	Dong Nguyen
ON Semiconductor	Adrien Gambino	Oscar Frietas	Nikhilesh Kamath
	Christian Klein	Brady McCoy	Andrw Yoo
Parade Technologies, Inc.	Jian Chen	Paul Xu	
	Craig Wiley	Alan Yuen	
Qualcomm, Inc	Amit Gupta	Will Kun	George Paparrizos
Realtek Semiconductor Corp.	Charlie Hsu	Ryan Lin	Terry Lin
	Ray Lee	Jay Lin	Changhung Wu
Renesas Electronics Corp.	Hajime Nozaki	Phillip Leung	Kiichi Muto
Richtek Technology Corporation	Kenny Chan	Bryan Huang	Roger Lo
	HM Chang	Chunan Kuo	Scott Wu
	Patrick Chang	Tony Lai	Alex Yang
Ricoh Company Ltd.	Yasuyuki Hayashi	Satoshi Oie	
	Tatsuya Irisawa	Yuuji Tsutsui	
ROHM Co., Ltd.	Kris Bahar	Nobutaka Itakura	Takashi Sato
	Ruben Balbuena	Yoshinori Ohwaki	
SiliConch Systems Private Limited	Jaswanth Ammineni	Aniket Mathad	Rakesh Polasa
	Pavitra Balasubramanian	Shubham Paliwal	Abhishek Sardeshpande
	Kaustubh Kumar		
Specwerkz	Bob Dunstan		
STMicroelectronics	Jérôme Bach	Christophe Cochard	Meriem Mersel
	Nathalie Ballot	Cedric Force	Federico Musarra
	Michel Bazin	Jessey Guilbot	Richard O'Connor
	Filipo Bonaccorso	Christophe Lorin	Legrand Pascal
Synopsys, Inc.	Zongyao Wen		
Texas Instruments	Felipe Balbi	Mike Campbell	Deric Waters
VIA Technologies, Inc.	Jay Tseng	Fong-Jim Wang	

Revision, Version History

Revision	Version	Date	Description
2.0	1.3	January 2022	Update Release Reprint release including incorporation of all approved ECNs as of the revision date plus editorial clean-up.
2.0	1.2	March 2020	Update Release Reprint release including incorporation of all approved ECNs as of the revision date plus editorial clean-up.
2.0	1.1	March 2020	Update Release Reprint release including incorporation of all approved ECNs as of the revision date plus editorial clean-up.
2.0	1.0	October 2017	Update Release Reprint release including incorporation of all approved ECNs as of the revision date plus editorial clean-up.
1.0	1.2	Nov 28, 2016	Update Release Reprint release including incorporation of all approved ECNs as of the revision date plus editorial clean-up.
1.0	1.1	July 2016	Update Release 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

1 Introduction

With the continued success of USB Power Delivery, there exists a need to define a common interface between a USB Type-C® Port Manager and 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*** Specification. The connection between the TCPM and the TCPC is defined as the USB Type-C Port Controller Interface, TCPCI.

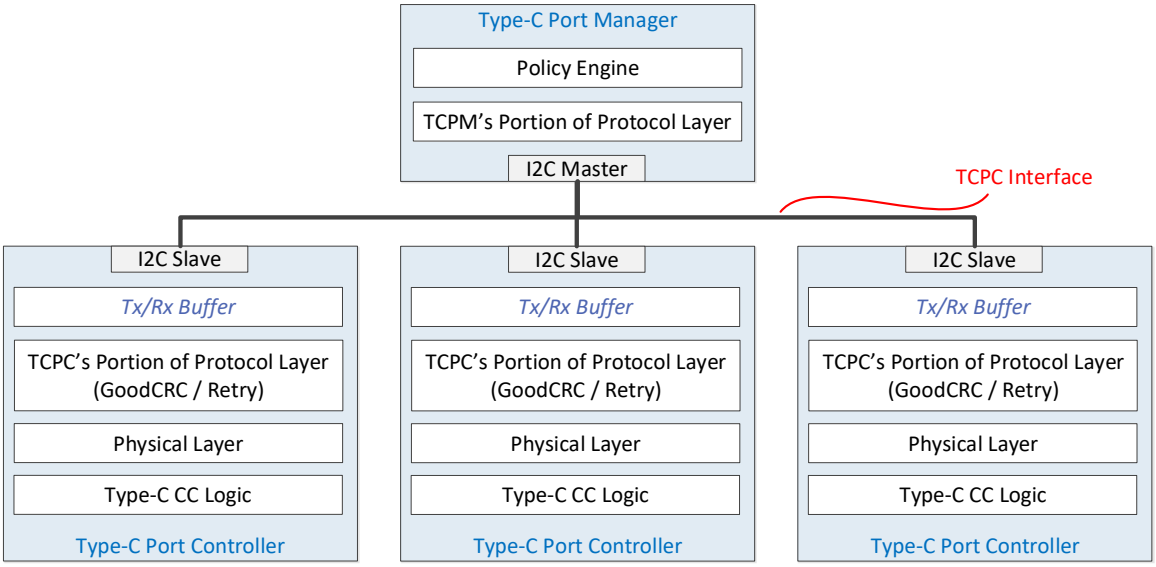


Figure 1-1. USB Type-C Port Manager to USB Type-C Port Controller Interface

1.1 Purpose

The USB Type-C Port Controller Interface (TCPCI) is the interface between a USB Type-C Port Manager (TCPM) and a USB Type-C Port Controller (TCPC). The goal of the TCPCI is to provide a defined interface between a TCPC and a TCPM in order to standardize and simplify TCPM implementations.
The TCPC is a functional block which encapsulates VBUS and VCONN power controls, ***USB Type-C*** CC logic, the ***USB PD*** BMC Physical Layer and portion of the ***USB PD*** Protocol Layer.

1.2 Scope

This specification is intended as a supplement to ***USB 3.2***, ***USB Type-C***, and ***USB PD*** specifications. It addresses only the elements required to implement and support the ***USB Type-C*** Port Controller.
Normative information is provided to allow interoperability of components designed to this specification. Informative information, when provided, may illustrate possible design implementations.

1.3 Related Documents

USB 3.2.	*Universal Serial Bus Revision 3.2 Specification* https://www.usb.org/documents
USB PD	*USB Power Delivery Specification, Revision 3.1, V1.3 January, 2022* https://www.usb.org/documents
USB Type-C	*USB Type-C Cable and Connector Specification, Revision 2.1, January, 2022* https://www.usb.org/documents

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.

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, per ***USB Type-C***
SNK.Open	Sink CC pin below maximum vRa, per ***USB Type-C***
SRC.Ra	Source CC pin above vOPEN, per ***USB Type-C***
SRC.Rd	Source CC pin within the vRd range, per ***USB Type-C***
SRC.Open	Source CC pin below maximum vRa, per ***USB Type-C***
OCP	Over-current Protection
OVP	Over-voltage Protection
vSafe0V	Safe operating voltage at “zero volts” per ***USB PD***
vSafe5v	Safe operating voltage at 5V per ***USB PD***

2 Overview

2.1 Introduction

2.2 USB Type-C Port Controller (TCPC) Interface

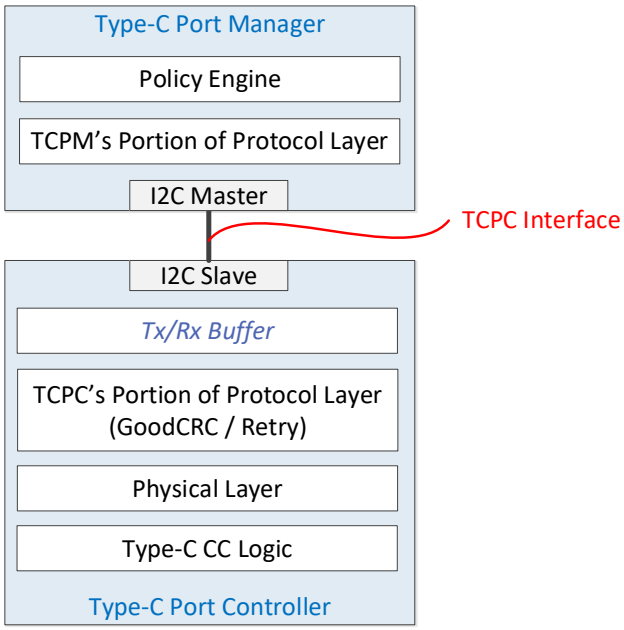


Figure 2-1. TCPC Interface

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*** CC logic, and the ***USB PD*** BMC physical layer and protocol layer other than the message creation. The TCPC shall not include support for USB PD BFSK.

2.3 Changes from Revision 1.0

The following is a summary of major changes between this specification (TCPCI Rev2 v1.2) and TCPCI Rev1 v1.2:

- Support for ***USB PD*** Extended Messages. The TRANSMIT_BUFFER and RECEIVE_BUFFER registers are redefined to accommodate 260 data bytes. Changes are made to the procedures of ***USB PD*** communication (see Sections 4.7).
- Support for ***USB PD*** Fast Role Swap (see Section 4.4.5.4.6).
- Support for SMBus PEC (Packet Error Checking) mechanism (see Sections 4.3.7 and 4.3.12).
- Support for a general-purpose timer (see Section 4.4.11).
- Support for vSafe0V reporting in EXTENDED_STATUS register (see Section 4.4.6.4).
- The TCPM is required to adopt parts of the SMBus protocol for reading and writing the I2C registers (see Section 4.3)
- Unless the TCPM sets TCPC_CONTROL.EnableLooking4ConnectionAlert bit, TCPC by default masks Alert assertion when CC_STATUS.Looking4Connection changes state.

- Support for optional normative way to set the target voltage for sourcing non-default voltages over VBUS (see Section 4.4.19).
- Added a TCPM timing requirement for servicing TCPC Alerts if the TCPM is connected to a TCPC Source which has set SinkTxOK to indicate to the Sink it is OK to send Atomic Message Sequences (AMS) (see Appendix C).
- Corrected the calculation example of automatic VBUS Sink discharge upon disconnect in Section 4.4.5.4.2.

2.4 Changes from Revision 2.0 Version 1.0

The following is a summary of major changes between this specification (TCPCI Rev2 v1.2) and TCPCI Rev2 v1.0:

- ALERT register is not reset upon a Hard Reset. (see Sections 4.4.2 and 4.7.3) Only the mask registers (ALERT_MASK, POWERE_STATUS_MASK, EXTENDED_STATUS_MASK, ALERT_EXTENDED_MASK) are reset upon a Hard Reset.
- Added two fields, namely DEVICE_CAPABILITIES_2.MessageDisableDisconnect and RECEIVE_DETECT.MessageDisableDisconnect, which allow a Sink TCPC to disable PD messaging based only on SNK.Open (Sink CC pin is below maximum vRa) is detected for at least tPDDebounce min (10ms). This allows supporting CTVPD CTUnattached.SNK state.
- A Source only TCPC is required to apply Rdch termination on CC pin that was providing VCONN until the CC pin reaches below vVCONNDischarge as specified in ***USB Type-C***. (see Sections 3.1 and 4.4.5.4).
- Added an unconditionally transition from Debug_Accessry_Exit to State_Debug_Accessory (as shown in Figure 4-21) to allow the Debug Accessory State-Machine to continue operating after a disconnect event.

2.5 Changes from Revision 2.0 Version 1.1

The following is a summary of major changes between this specification (TCPCI Rev2 v1.2) and TCPCI Rev2 v1.1:

- Clarified that COMMAND.Look4Connection will only cause the TCPC to toggle DRP if POWER_CONTROL.AutoDischargeDisconnect = 0. The appropriate way for the TCPM to start a DRP toggling is to clear POWER_CONTROL.AutoDischargeDisconnect before writing COMMAND.Look4Connection.

2.6 Changes from Revision 2.0 Version 1.2

The following is a summary of major changes between this specification (TCPCI Rev2 v1.3) and TCPCI Rev2 v1.2:

- Added support for the Extended Power Range (EPR).
    - Modified Table 4-41 for the VBUS_SINK_DISCONNECT_THRESHOLD Register to increase number of bits for the Voltage trip
    - Similarly, modified table 4-42 for the VBUS_STOP_DISCHARGE_THRESHOLD Register Description Table 4-43 for the VBUS_VOLTAGE_ALARM_HI_CFG and VBUS_VOLTAGE_ALARM_LO_CFG Register Descriptions
    - Table 3-2 Required DEVICE_CAPABILITY_2 Support added B15 for the VBUS Voltage Format and bits 5:4 clarification

- Added DEVICE_CAPABILITIES_3 Register definition as an Optional register in case TCPC supports EPR voltage ranges.

3 USB 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*** port power control for VBUS and VCONN (normative)
- ***USB Type-C*** CC control and sensing (normative)
- ***USB PD*** Message delivery (normative)

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 the registers required to allow the TCPM to control VBUS Sourcing. A Sink capable TCPC shall provide the registers to allow the TCPM to Control VBUS Sinking.
To ensure safety in case the I2C interface fails, a TCPC sourcing nondefault VBUS voltage (other than vSafe5V) 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 be implemented to discharge VBUS. Force discharge is a larger current discharge used to discharge VBUS to below vSafe0V upon detecting a disconnect per ***USB Type-C*** (exiting the Attached_Src state in Figure 4-19 and Figure 4-20).
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 nondefault voltages (i.e. voltages other than vSafe5V).
A Source Only or DRP TCPC shall include control for VCONN sourcing. A Sink Only TCPC shall include control for VCONN sourcing if VCONN Swap or Sink with Accessory is supported. VCONN sourcing shall meet requirements as provided in the ***USB Type-C***, including tVCONNON and tVCONNOFF parameters. When a Source Only TCPC is requested to disable VCONN sourcing (by setting POWER_CONTROL.EnableVCONN=0b), the Source Only TCPC shall apply Rdch termination on the CC pin that was providing VCONN until the CC pin is discharged below vVCONNDischarge level as specified in the ***USB Type-C***. A DRP TCPC may not provide the capability of applying Rdch termination when disabling VCONN sourcing is requested.
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 the normative 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*** BFSK. The TCPC shall implement the portion of the Protocol Layer in the ***USB PD*** specification as shown in Figure 3-3, Figure 3-4, and Figure 3-5. The TCPC is opaque from a ***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 ***USB PD*** messages.
The TCPC shall implement the entire ***USB PD*** Physical Layer with BMC encoding. The TCPC shall implement the following portions of the ***USB PD*** Protocol Layer:

- 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
- A message transmission is considered successful after receiving a GoodCRC response with the matching MessageID and SOP type
- 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 TransmitSOP*MessageDiscarded bit in the ALERT register shall be 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*** 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.

3.4 Debug Accessory Detection (Optional Normative)

The TCPC may implement autonomous detection of the Debug Accessory State (vRd/vRd) per ***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.
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*** specification.

3.5 State Diagram Introduction

The TCPC state diagrams defined in this specification are normative and shall define the operation of the TCPC. Note that TCPC 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.

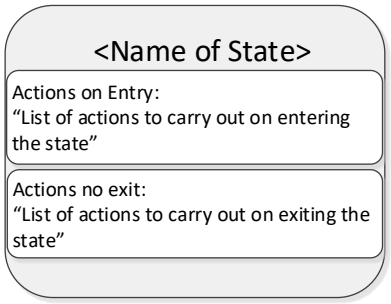


Figure 3-1. Outline of States

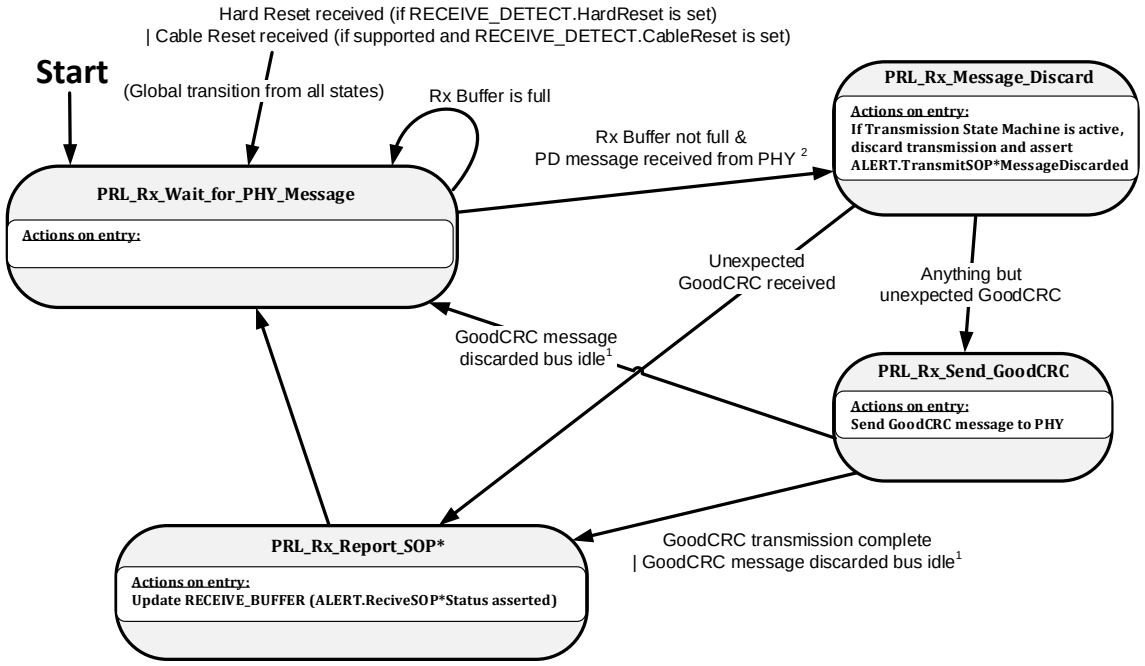
Transitions from one state to another are indicated by arrows with the conditions listed on the arrow. Where there are multiple conditions these are connected using either a logical OR “|” or a logical AND “&”. The inverse of a condition is shown with a “NOT” in front of the condition. In some cases, 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

3.6 TCPC Protocol Layer State Operation

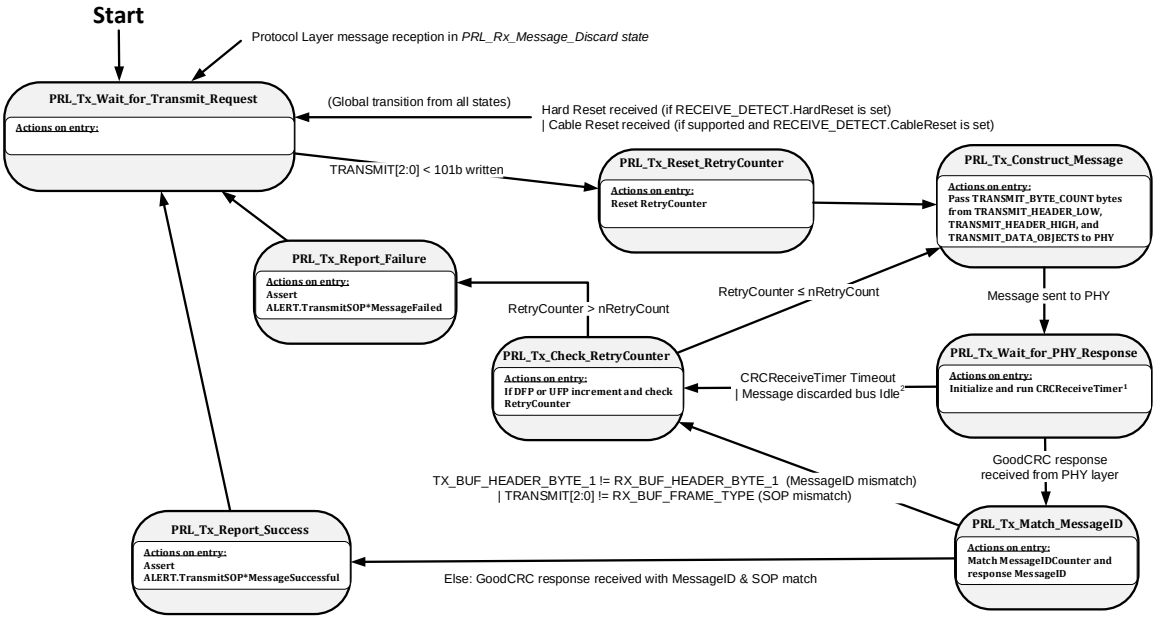
This section describes the normative TCPC Protocol Layer state operation. The informative TCPM Protocol Layer state operation can be found in Appendix A.
The state operation as shown in Figure 3-3, Figure 3-4 and Figure 3-5 depends on the message detection settings in RECEIVE_DETECT register (as provided in Section 4.4.13). For example, if RECEIVE_DETECT.EnableSOP''Message is not set, the TCPC does not transition from *PRL_Rx_Wait_for_PHY_Message* to *PRL_Rx_Message_Discard* when it receives SOP'' message regardless of whether the RECEIVE_BUFFER is full. Similarly, if RECEIVE_DETECT.CableReset is not set, the TCPC does not perform global transition from all states to *PRL_Rx_Wait_for_PHY_Message*, *PRL_Tx_Wait_for_Transmit_Request* or *PRL_HR_Wait_for_Hard_Reset_Request* state when it receives Cable Reset.

Figure 3-3 shows that when the TCPC receives an unexpected GoodCRC message, the TCPC sets ReceiveSOP*MessageStatus bit in the ALERT register to allow the TCPM reading the unexpected GoodCRC message as a regular USB PD message (as described in Section 4.7.5). The TCPM should follow ***USB PD*** if the port receives an unexpected GoodCRC message. On the other hand, if the TCPC receives an expected GoodCRC message (that contains the matching MessageID and SOP type with the transmitted PD message), then the TCPC would enter *PRL_Tx_Report_Success* TCPC Protocol Layer state (as shown in Figure 3-4) and set TransmitSOP*MessageSuccessful bit in the ALERT register.



[1] This transition is taken by the TCPC 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 message is only expected after the TCPC has sent a PD message, and the TCPC Protocol Layer State Machine is in *PRL_Tx_Wait_for_PHY_Response*).

Figure 3-3. Message Reception State Diagram implemented in TCPC



[1] The CRCReceiveTimer is only started after the TCPC 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 specification).
[2] This indication is sent by the PHY Layer when a message has been discarded due to CC being busy, and after CC becomes idle again (see USB-PD specification). The CRCReceiveTimer is not running in this case since no message has been sent.

Figure 3-4. Message Transmission State Diagram Implemented in TCPC

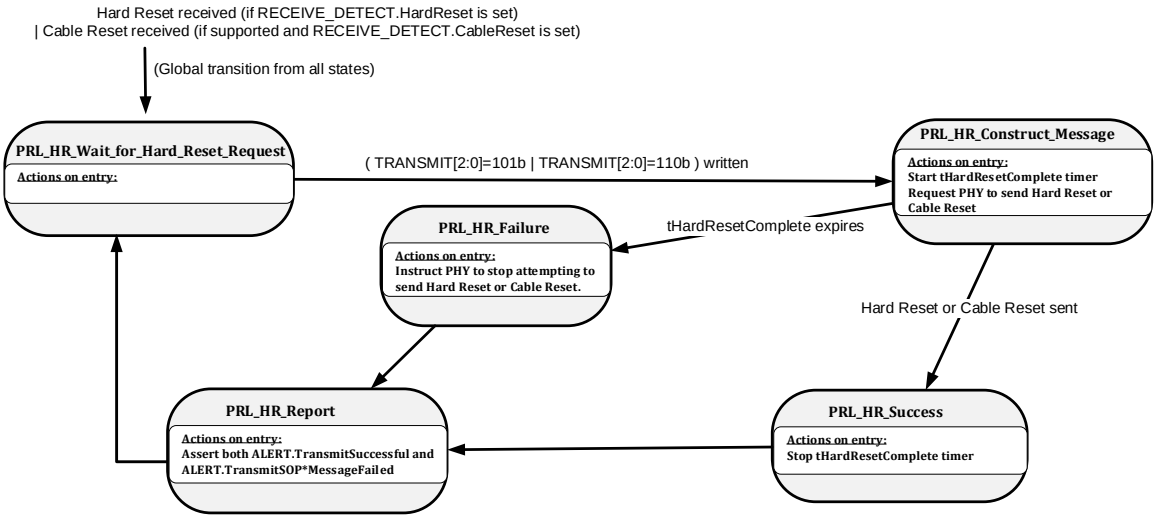


Figure 3-5. Hard Reset Transmission State Diagram Implemented in TCPC

3.7 Source, Sink, and DRP Requirements

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.

Table 3-1. Required DEVICE_CAPABILITIES_1 Support

Power Role		B15 VBUS Voltage Target	B14 OCP	B13 OVP	B12 Bleed Discharge	B11 Force Discharge	B10 VBUS Alarm Meas	B9…8 Rp Value	B7…5 Power Roles	B4 SOP'/ SOP" DBG	B3 Source VCONN	B2 Sink VBUS	B1 Source Non-default	B0 Source VBUS
Source-only (Nondefault VBUS)		O	O[2]	O[3]	O	R	R	01b 10b	001b	O	R	O	R	R
Source-only (Default VBUS)		O	O[2]	O[3]	O	R	O	00b	001b	O	R	O	O	R
Sink-only (Nondefault VBUS)		O	O	O[3]	O	O	R	00b	010b	O	O	R	O	O
Sink-only (Default VBUS)		O	O	O[3]	O	O	O	00b	010b	O	O	R	O	O
DRP	Toggling (Source/Sink) (Nondefault VBUS)	O	O[2]	O[3]	O	R	R	01b 10b	100b 101b 110b	O	R	R	R	R
	Toggling (Source/Sink) (Default VBUS)	O	O[2]	O[3]	O	R	O	00b	100b 101b 110b	O	R	R	O	R
	Sourcing Device (Nondefault VBUS)	O	O[2]	O[3]	O	R	R	01b 10b	100b 101b 110b	O	R	O	R	R
	Sourcing Device (Default VBUS)	O	O[2]	O[3]	O	R	O	00b	100b 101b 110b	O	R	O	O	R
	Sinking Host (Nondefault VBUS)	O	O	O[3]	O	O	R	00b	100b 101b 110b	O	O	R	O	O
	Sinking Host (Default VBUS)	O	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 Overcurrent Fault. If this bit is not set, then the OCP event is not visible to TCPC.
3. Device_Capabilities_1.VbusOVPReporting (B13) defines if reporting of the OVP event is supported or not. OVP is required per USB-PD.

Table 3-2. Required DEVICE_CAPABILITIES_2 Support

Power Role		B15 DEVICE CAPABILITY 3	B14 Message Disable Disconnect	B13 Generic Timer	B12 Long Message	B11 SM Bus PEC	B10 Source FR Swap	B9 Sink FR Swap	B8 Watchdog Timer	B7 Sink Disconnect Detection	B6 Stop Discharge Threshold	B5…4 VBUS Alarm	B3…1 VCONN Power	B0 VCONN Over Current Fault
Source-only (Nondefault VBUS)		0	0	0	0[2]	0	0	0	0	0	R	0	000b or other	0
Source-only (Default VBUS)		0	0	0	0	0	0	0	0	0	0	0	000b or other	0
Sink-only (Nondefault VBUS)		0	0	0	0[2]	0	0	0	0	R	0	0	X	0
Sink-only (Default VBUS)		0	0	0	0	0	0	0	0	0	0	0	X	0
DRP	Toggling (Source/Sink) (Nondefault VBUS)	0	0	0	0[2]	0	0	0	0	R	R	0	000b or other	0
	Toggling (Source/Sink) (Default VBUS)	0	0	0	0	0	0	0	0	0	0	0	000b or other	0
	Sourcing Device (Nondefault VBUS)	0	0	0	0[2]	0	0	0	0	0	R	0	000b or other	0
	Sourcing Device (Default VBUS)	0	0	0	0	0	0	0	0	0	0	0	000b or other	0
	Sinking Host (Nondefault VBUS)	0	0	0	0[2]	0	0	0	0	R	0	0	000b or other	0
	Sinking Host	0	0	0	0	0	0	0	0	0	0	0	000b or other	0

	(Default V_{BUS})													

Notes:

1. R=Required, O=Optional and X=Don't-care
2. Required if Extended Messages longer than 30 Bytes long (such as EPR_Source_Capabilities or EPR_Sink_Capabilities) is supported and chunking is not supported. In this case, DEVICE_CAPABILITES_2.LongMessages shall be set to 1b.

3.7.1 Source Requirements

A TCPC, which supports Source operation, shall implement the following:

1. Provide control of VBUS source path (see COMMAND register, Section 4.4.8).
2. Optionally 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. Provide control of a VCONN switch (see POWER_CONTROL.VCONNPowerSupported and POWER_CONTROL.EnableVconn).
4. Optionally include monitoring for the presence of VCONN (see POWER_STATUS.VCONNPresent).
5. Support Device_Capabilities_1 and Device_Capabilities_2 register for the Source-only (Nondefault VBUS) or Source-only (Default VBUS) Power Role as defined in Table 3-1 and Table 3-2.

Table 3-3. Source Requirements

Name	Functionality
USB-PD	
VCONN Swap	Optional
Power Role Swap Support	Optional
Fast Role Swap Support	Optional
USB-PD Extended Message Support	Optional, but Required if EPR is supported
CC CONTROL	
CC Detect Status	Required
Port Disable	Required (Rp to zOpen)
Power Roles Supported	SRC (Rp default, 1.5A, 3A) indicated in DEVICE_CAPABILITIES_1.SourceResistorSupported SNK (Rd) Optional
PORT POWER CONTROL	
Power Status	Required
Supply VCONN	Required
Sink VBUS	Optional
Supply VBUS	Required
Dead Battery	Optional Required if DRP has a battery (present Rd when no power)

3.7.2 Sink Requirements:

A TCPC, which supports Sink operation, shall implement the following:

1. Contain CC logic that implements a mechanism to present Rd in a dead battery condition (see Table 4-17. Power On Default Conditions).
2. Optionally include the monitoring of the presence of VCONN (see POWER_CONTROL.VCONNPowerSupported and POWER_STATUS.VCONNPresent).
3. Provide control of VBUS sink path (see COMMAND register, Section 4.4.8).
4. Provide a mechanism for detecting a disconnect if it is capable of sinking a voltage other than vSafe5V (see Section 4.4.18.1).
5. Provide a mechanism for detecting vSafe0V.
6. Support Device_Capabilities_1 and Device_Capabilities_2 register for the Sink-only (Nondefault VBUS) or Sink-only (Default VBUS) Power Role as defined in Table 3-1 and Table 3-2.

Table 3-4. Sink Requirements

Name	Functionality
USB-PD	
VCONN Swap	Optional
Power Role Swap Support	Optional
Fast Role Swap Support	Optional
USB-PD Extended Message Support	Optional, but Required if EPR is supported
CC CONTROL	
CC Detect Status	Required
Port Disable	Required (Rd to zOpen)
Power Roles Supported	SNK (Rd) Required 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)

3.7.3 Sink with Accessory Support

A TCPC, which supports Sink with Accessory Support operation, shall implement the following:

1. Contain CC logic that implements a mechanism to present Rd in a dead battery condition (see Table 4-17. Power On Default Conditions).
2. Provide control of VCONN source path (see POWER_CONTROL.VCONNPowerSupported and POWER_CONTROL.EnableVCONN).
3. Optionally include the monitoring of the presence of VCONN (see POWER_STATUS.VCONNPresent).
4. Provide control of VBUS sink path (see COMMAND register, Section 4.4.8).
5. Provide a mechanism for detecting a disconnect if it is capable of sinking a voltage other than vSafe5V (see Section 4.4.18.1).
6. Provide a mechanism for detecting vSafe0V.
7. Support Device_Capabilities_1 and Device_Capabilities_2 register for the Sink-only (Nondefault VBUS) or Sink-only (Default VBUS) Power Role as defined in Table 3-1 and Table 3-2.

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

Table 3-5. Sink with Accessory Support Requirements

Name	Functionality
USB-PD	
VCONN Swap	Required
Power Role Swap Support	Optional
Fast Role Swap Support	Optional
USB-PD Extended Message Support	Optional, but Required if EPR is supported
CC CONTROL	
CC Detect Status	Required
Port Disable	Required (Rp to zOpen)
Power Roles Supported	SNK (Rd) Required 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)

3.7.4 DRP Requirements

A TCPC, which supports Dual Role Port operation, shall implement the following:

1. Contain CC logic to detect the insertion of a Source, Sink, and Audio and debug accessory (see ROLE_CONTROL).
2. Contain CC logic that implements a mechanism to present Rd in a dead battery condition (see CC_STATUS).
3. Provide control of VBUS source path (see COMMAND register, Section 4.4.8).
4. Provide control for a VCONN switch (see POWER_CONTROL.VCONNPowerSupported and POWER_CONTROL.EnableVCONN)
5. Include the monitoring of the presence of VCONN (see POWER_STATUS.VCONNPresent).
6. Provide a mechanism for detecting a disconnect if it is capable of sinking a voltage other than vSafe5V (see Section 4.4.18.1).
7. Provide a mechanism for detecting vSafe0V.
8. Support Device_Capabilities_1 and Device_Capabilities_2 register for at least the DRP Toggling, Sourcing Device, and Sinking Host (Default VBUS) Power Roles as defined in Table 3-1 and Table 3-2.

Table 3-6. DRP Requirements

Name	Functionality
USB-PD	
VCONN Swap	Optional
PR Swap Support	Optional
Fast Role Swap Support	Optional
USB-PD Extended Message Support	Optional, but Required if EPR is supported
CC CONTROL	
CC Detect Status	Required
Port Disable	Required (Rp to zOpen)
Power Roles Supported	SRC (Rp default, 1.5A, 3A) indicated in DEVICE_CAPABILITIES_1.SourceResistorSupported SNK (Rd) Required
PORT POWER CONTROL	
Power Status	Required
Supply VCONN	Required
Sink VBUS	Required
Supply VBUS	Required
Dead Battery	Required (present Rd when no power)

3.8 Watchdog Timer Requirements (Optional Normative)

It is recommended a watchdog timer, which monitors the TCPM-to-TCPC interface for lack of communication, be implemented by a TCPC if it supports sourcing or sinking nondefault voltages (i.e. voltages other than vSafe5V). A watchdog timer shall be implemented when 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.

3.8.1 Watchdog Timer Function

When enabled, the watchdog timer shall start when the Alert# pin is asserted.
An unresponsive TCPM which is unable to clear the interrupt within tHVWatchdog, Table 4-49, 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 the Sink paths, discharge V_{BUS} to vSafe0V, and then set FAULT_STATUS.I2CInterfaceError. The TCPC shall remove the V_{BUS} discharge circuit when V_{BUS} is below vSafe0V and it shall not re-apply the discharge circuit if V_{BUS} rises above vSafe0V. Stop discharge in this case is an edge-triggered event.
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.

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 logic and ***USB PD*** message delivery through a simple register interface. The normative 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 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
4. The TCPM designer must meet the I2C bus loading requirements when determining the maximum number of devices on the I2C bus.
5. 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.
6. The TCPC shall support Fast-mode Plus (Fm+) bus speed. It may also support other bus speeds.
7. The TCPC shall have an open drain active low output Alert# pin. This pin is used to indicate a change of state, where Alert# pin is asserted when any Alert Bits are set.
8. The TCPCI shall support an I/O voltage range from 1.8V to 3.6V.
9. The TCPC shall allow reads to every register even when it is defined as Write only. The TCPM should assume the register information returned from a Write only register is not valid.
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 TCPCI adopts parts of the SMBus protocol as normative requirements. Each register shall be accessed by reading or writing at the first byte in the register. Section 4.3 provides the normative way for the TCPM to read and write the registers. The TCPC may implement the following protocol:
    - If the TCPM reads a register address that is not the first byte in that register, the TCPC may assert FAULT_STATUS.I2CInterfaceError and leave the SDA line open so the TCPM reads all 1's.
    - If the TCPM writes a register address that is not the first byte in that register, the TCPC may assert FAULT_STATUS.I2CInterfaceError and ignore the write.
    - If the TCPM reads multiple registers in a single I2C transaction, the TCPC may assert FAULT_STATUS.I2CInterfaceError and leave the SDA line open so the TCPM reads all 1's.
    - If the TCPM writes multiple registers in a single I2C transaction, the TCPC may assert FAULT_STATUS.I2CInterfaceError and ignore the write.

One of the following three actions is allowed if TCPM writes to a register or a bit that is not implemented or that is reserved:

- The TCPC ignores it and does nothing
- The TCPC does nothing except generating a bit-level Not Acknowledge signal (a NAK where SDA remains HIGH during the ninth clock pulse)
- The TCPC does nothing except asserting FAULT_STATUS.I2CInterfaceError

4.1 SMBus with Packet Error Checking Mechanism (Optional Normative)

Some TCPCs may implement the PEC (Packet Error Checking) mechanism to improve the I2C communication robustness. When the Packet Error Checking mechanism is enabled (i.e. TCPC_CONTROL.EnableSMBusPEC = 1b), each transaction shall be appended by a Packet Error Code (PEC) byte. The PEC calculation uses CRC-8 as defined in the SMBus specification (Section 6.4 of the SMBus Specification, version 3.0 available at http://smbus.org/specs/). If

TCPC_CONTROL.EnableSMBusPEC = 1b, the TCPC shall check the validity of the PEC in real time when the TCPM writes to the TCPC. If an incorrect PEC is discovered in the write transaction, the TCPC shall generate a bit-level NAK to the PEC byte.

4.2 Register Map

The 16-bit (2-byte) 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. Refer to Sections 4.3.3, 4.3.4, 4.3.9 and 4.3.10 for details on how to read/write 2-byte registers.

Table 4-1. Register Map

Address	Register Name	Normative /Optional?	Type	Reset Value	Definition
00h…01h	VENDOR_ID	Normative	R	VD	Table 4-2. VENDOR_ID Register Definition
02h…03h	PRODUCT_ID	Normative	R	VD	Table 4-3. PRODUCT_ID Register Definition
04h…05h	DEVICE_ID	Normative	R	VD	Table 4-4. DEVICE_ID Register Definition
06h….07h	USBTYPEC_REV	Normative	R	VD	Table 4-5. USBTYPEC_REV Register Definition
08h…09h	USBPD_REV_VER	Normative	R	VD	Table 4-6. USBPD_REV_VER Register Description
0Ah…0Bh	PD_INTERFACE_REV	Normative	R	VD	Table 4-7. PD_INTERFACE_REV Register Description
0Ch…0Fh	Reserved	Normative			Intentionally Blank
10h…11h	ALERT	Normative	R/W	0000h	Table 4-8. ALERT Register Definition
12h…13h	ALERT_MASK	Normative	R/W	7FFFh	Table 4-9. ALERT_MASK Register Definition
14h	POWER_STATUS_MASK	Normative	R/W	FFh	Table 4-10. POWER_STATUS_MASK Register Definition
15h	FAULT_STATUS_MASK	Normative	R/W	FFh	Table 4-11. FAULT_STATUS_MASK Register Definition
16h	EXTENDED_STATUS_MASK	Normative	R/W	01h	Table 4-12. EXTENDED_STATUS_MASK Register Definition
17h	ALERT_EXTENDED_MASK	Normative	R/W	07h	Table 4-13. ALERT_EXTENDED_MASK Register Definition
18h	CONFIG_STANDARD_OUTPUT	Optional	R/W	60h	Table 4-14. CONFIG_STANDARD_OUTPUT Register Definition
19h	TCPC_CONTROL	Normative	R/W	00h	Table 4-15. TCPC_CONTROL Register Definition
1Ah	ROLE_CONTROL	Normative	R/W	Table 4-17	Table 4-16. ROLE_CONTROL Register Definition
1Bh	FAULT_CONTROL	Partial Normative	R/W	00h	Table 4-18. FAULT_CONTROL Register Definition
1Ch	POWER_CONTROL	Partial Normative	R/W	60h	Table 4-19. POWER_CONTROL Register Definition
1Dh	CC_STATUS	Normative	R		Table 4-22. CC_STATUS Register Definition
1Eh	POWER_ STATUS	Normative	R		Table 4-23. POWER_STATUS Register Definition
1Fh	FAULT_STATUS	Normative	R/W	80h	Table 4-24. FAULT_STATUS Register Definition

Address	Register Name	Normative /Optional?	Type	Reset Value	Definition
20h	EXTENDED_STATUS	Normative	R		Table 4-25. EXTENDED_STATUS Register Definition
21h	ALERT_EXTENDED	Normative	R/W	00h	Table 4-26. ALERT_EXTENDED Register Description
22h	Reserved	Normative	R		Intentionally Blank
23h	COMMAND	Normative	W	00h	Table 4-27. COMMAND Register Definition
24h…25h	DEVICE_CAPABILITIES_1	Normative	R	VD	Table 4-28. DEVICE_CAPABILITIES_1 Register Definition
26h…27h	DEVICE_CAPABILITIES_2	Normative	R	VD	Table 4-29. DEVICE_CAPABILITIES_2 Register Definition
28h	STANDARD_INPUT_CAPABILITIES	Normative	R	VD	Table 4-31. STANDARD_INPUT_CAPABILITIES Register Definition
29h	STANDARD_OUTPUT_CAPABILITIES	Normative	R	VD	Table 4-32. STANDARD_OUTPUT_CAPABILITIES Register Definition
2Ah	CONFIG_EXTENDED1	Normative	R/W	00h	Table 4-33. CONFIG_EXTENDED1 Register Definition
2Bh	Reserved	Normative	R		Intentionally Blank
2Ch…2Dh	GENERIC_TIMER	Optional	W	0000h	Table 4-34. GENERIC_TIMER Register Definition
2Eh	MESSAGE_HEADER_INFO	Normative	R/W	Table 4-17	Table 4-35. MESSAGE_HEADER_INFO Register Definition
2Fh	RECEIVE_DETECT	Normative	R/W	00h	Table 4-36. RECEIVE_DETECT Register Definition
RECEIVER_BUFFER (Normative) (Always Read at Address 30h)					
30h	READABLE_BYTE_COUNT	Normative	R	00h	Indicates the number of bytes in the RX_BUF_BYTE_x registers plus one (for the RX_BUF_FRAME_TYPE) Table 4-37. The content of this register is undefined when the RECEIVE_BUFFER is cleared.
	RX_BUF_FRAME_TYPE	Normative	R	00h	Type of received frame (Table 4-38). This register is “hidden” and can only be accessed by reading at address 30h.
	RX_BUF_BYTE_x	Normative	R	00h	Receive Buffer Bytes. These registers are “hidden” and can only be accessed by reading at address 30h.
50h	TRANSMIT	Normative	R/W	00h	Table 4-39. TRANSMIT Register Definition Transmit aggregate of data written to TRANSMIT_BUFFER since the pointer was last reset
TRANSMIT_BUFFER (Always Write at Address 51h)					
51h	I2C_WRITE_BYTE_COUNT	Normative	W	00h	Table 4-40. The number of bytes the TCPM writes to the TX_BUF_BYTEx in the given I2C/SMBus transaction. The TCPM shall write as many bytes in the buffer as defined in this register in one I2C write transaction.
	TX_BUF_BYTE_x	Normative	W	00h	Transmit Buffer Bytes. These registers are “hidden” and can only be accessed by writing to address 51h.

Address	Register Name	Normative /Optional?	Type	Reset Value	Definition
70h…71h	VBUS_VOLTAGE	Normative if nondefault V_{BUS}	R	0000h	Table 4-41. VBUS_VOLTAGE Register Definition
72h..,73h	VBUS_SINK_DISCONNECT_THRESHOLD	Normative if Sink nondefault V_{BUS}	R/W	008Ch (3.5V)	Table 4-42. VBUS_SINK_DISCONNECT_THRESHOLD Register Description
74h…75h	VBUS_STOP_DISCHARGE_THRESHOLD	Normative if Sink nondefault V_{BUS}	R/W	0020h (0.8V)	Table 4-43. VBUS_STOP_DISCHARGE_THRESHOLD Register Description
76h…77h	VBUS_VOLTAGE_ALARM_HI_CFG	Normative if nondefault V_{BUS}	R/W	0000h	Table 4-44. VBUS_VOLTAGE_ALARM_HI_CFG Register Description
78h…79h	VBUS_VOLTAGE_ALARM_LO_CFG	Normative if nondefault V_{BUS}	R/W	0000h	Table 4-45. VBUS_VOLTAGE_ALARM_LO_CFG Register Description
7Ah…7Bh	VBUS_NONDEFAULT_TARGET	Optional	R/W	0000h	Table 4-46. VBUS_NONDEFAULT_TARGET Register Description
7Ch..7Dh	DEVICE_CAPABILITIES_3	Optional	R	VD	Table 4-30. DEVICE_CAPABILITIES_3 Register Definition
7Eh…7Fh	Reserved	Normative			Intentionally Blank
80h…FFh	Vendor Defined Registers	Optional		VD	As many as Vendor Defines

Notes:
VD - Vendor Defined

4.3 Writing and Reading Registers

This section defines the protocol for the TCPM to read and write the I2C registers. The TCPCI adopts part of the SMBus protocol and this 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

The TCPC may implement the following protocol:

- If the TCPM reads a register address that is not the first byte in that register, the TCPC may assert FAULT_STATUS.I2CInterfaceError and leave the SDA line open so the TCPM reads all 1's.
- If the TCPM writes a register address that is not the first byte in that register, the TCPC may assert FAULT_STATUS.I2CInterfaceError and ignore the write.
- If the TCPM reads multiple registers in a single I2C transaction, the TCPC may assert FAULT_STATUS.I2CInterfaceError and leave the SDA line open so the TCPM reads all 1's.
- If the TCPM writes multiple registers in a single I2C transaction, the TCPC may assert FAULT_STATUS.I2CInterfaceError and ignore the write.

4.3.1 Writing Single Byte Registers

The TCPM cannot use the I2C short-cut to write consecutive registers in a single operation. For example: the TCPM shall use two transactions to write ROLE_CONTROL and FAULT_CONTROL as shown in Figure 4-1.

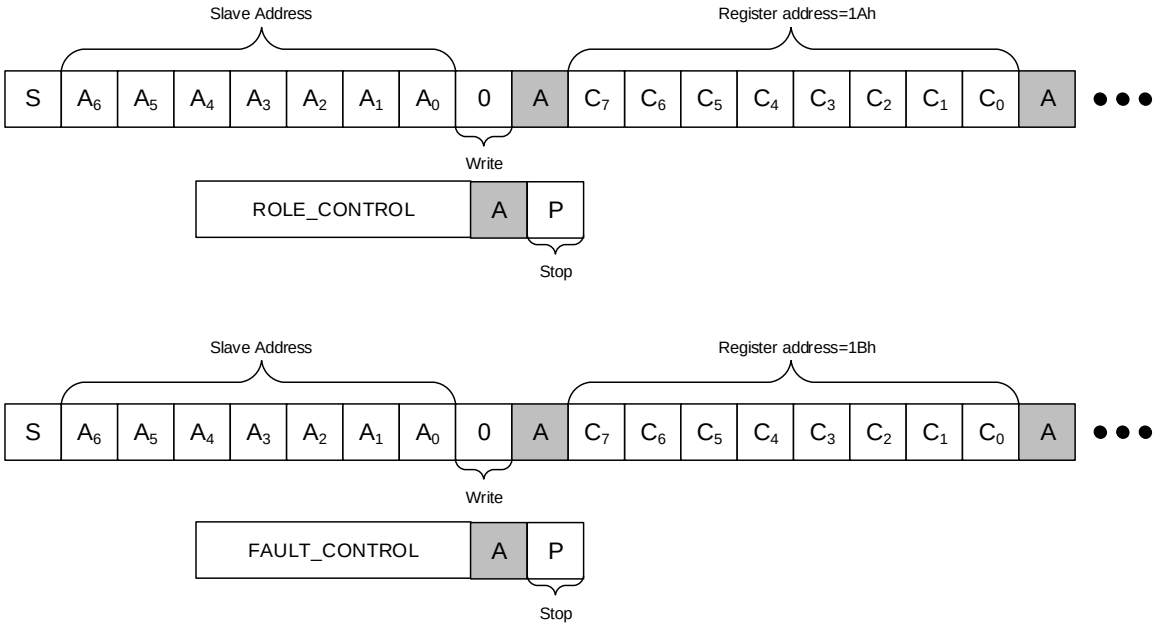


Figure 4-1. Writing Consecutive Single Byte Registers with or without the SMBUS Protocol

4.3.2 Reading Single Byte Registers

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

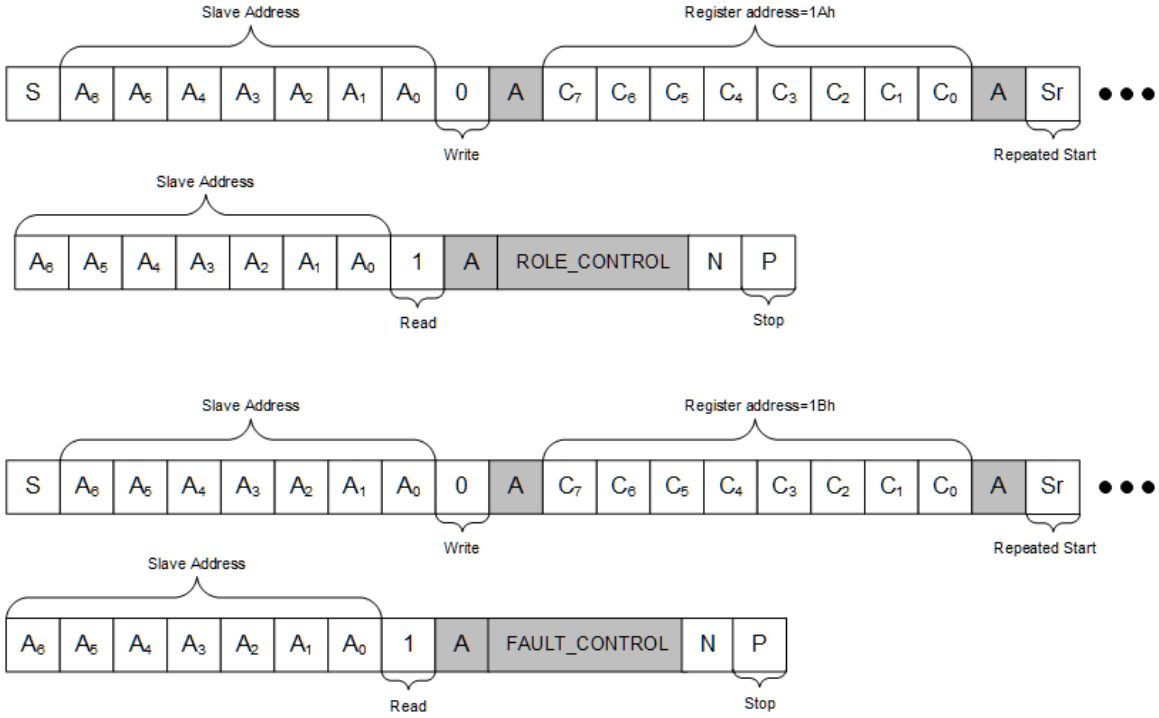


Figure 4-2. Reading Consecutive Single Byte Registers with or without the SMBus Protocol

4.3.3 Writing Two-Byte Registers

The TCPM shall write 2-byte register in a single I2C transaction. For example: the TCPM shall write both bytes in the ALERT register at the same time as depicted in Figure 4-3.

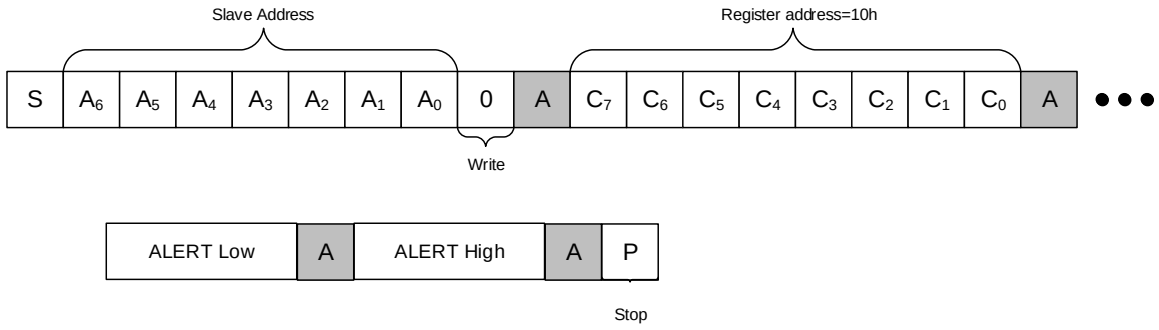


Figure 4-3. Writing a 2-Byte Register with or without the SMBus Protocol

4.3.4 Reading Two-Byte Registers

The TCPM shall read 2-byte register in a single I2C transaction. The TCPM shall read both bytes in the VENDOR_ID register at the same time as depicted in the following Figure 4-4.

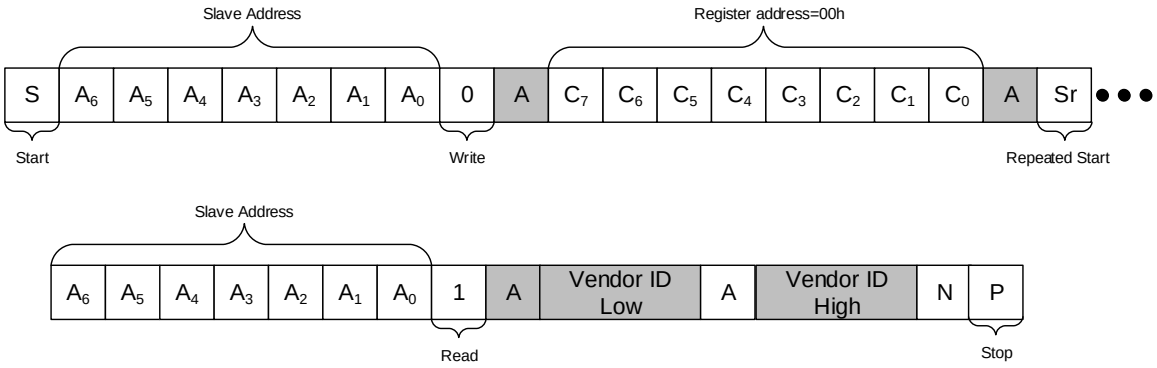


Figure 4-4. Reading a 2-Byte Register with or without the SMBus Protocol

4.3.5 Writing the TRANSMIT_BUFFER

Figure 4-5 illustrates how the transmit buffer shall be written with I2C or SMBus protocol.

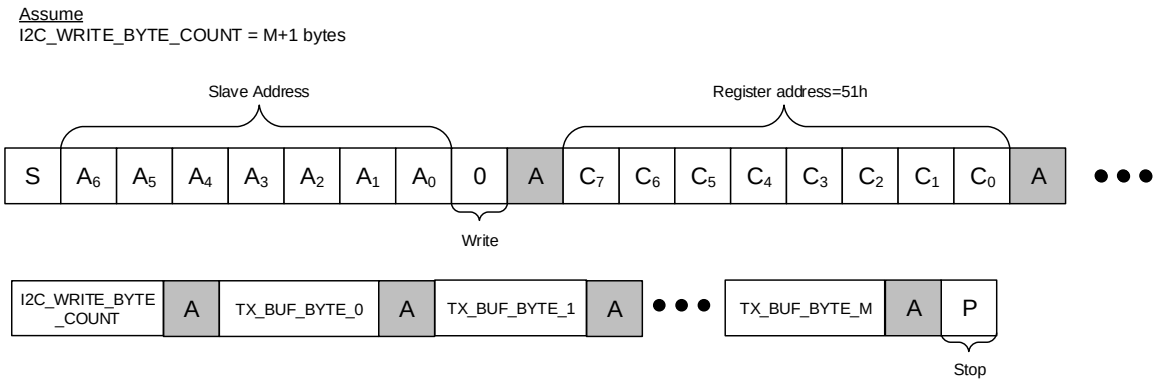


Figure 4-5. Writing the TRANSMIT_BUFFER with or without the SMBus Protocol

4.3.6 Reading the RECEIVE_BUFFER

Figure 4-6 illustrates how the receive buffer shall be read with I2C or SMBus protocol.

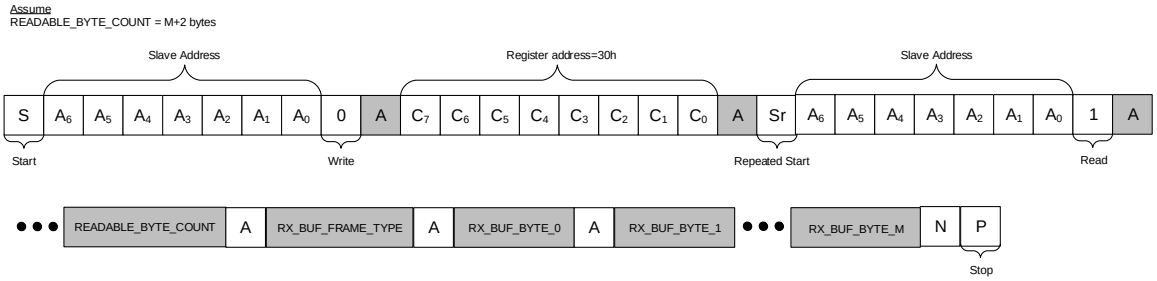


Figure 4-6. Reading the RECEIVE_BUFFER with or without the SMBus Protocol

4.3.7 Writing Single Byte Registers using SMBus with PEC

Figure 4-7 illustrates how the ROLE_CONTROL and FAULT_CONTROL can be written in two transactions using SMBus with PEC.

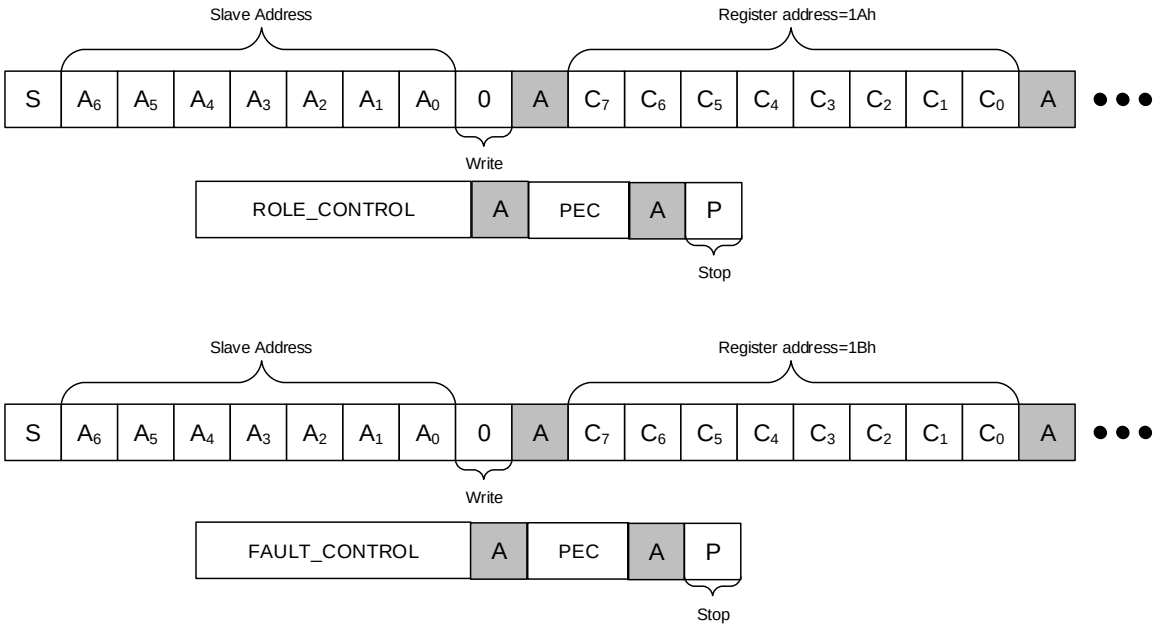


Figure 4-7. Writing Consecutive Single Byte Registers using SMBus PEC

4.3.8 Reading Single Byte Registers using SMBus with PEC

Figure 4-8 illustrates how the ROLE_CONTROL and FAULT_CONTROL can be read in two transactions using SMBus with PEC

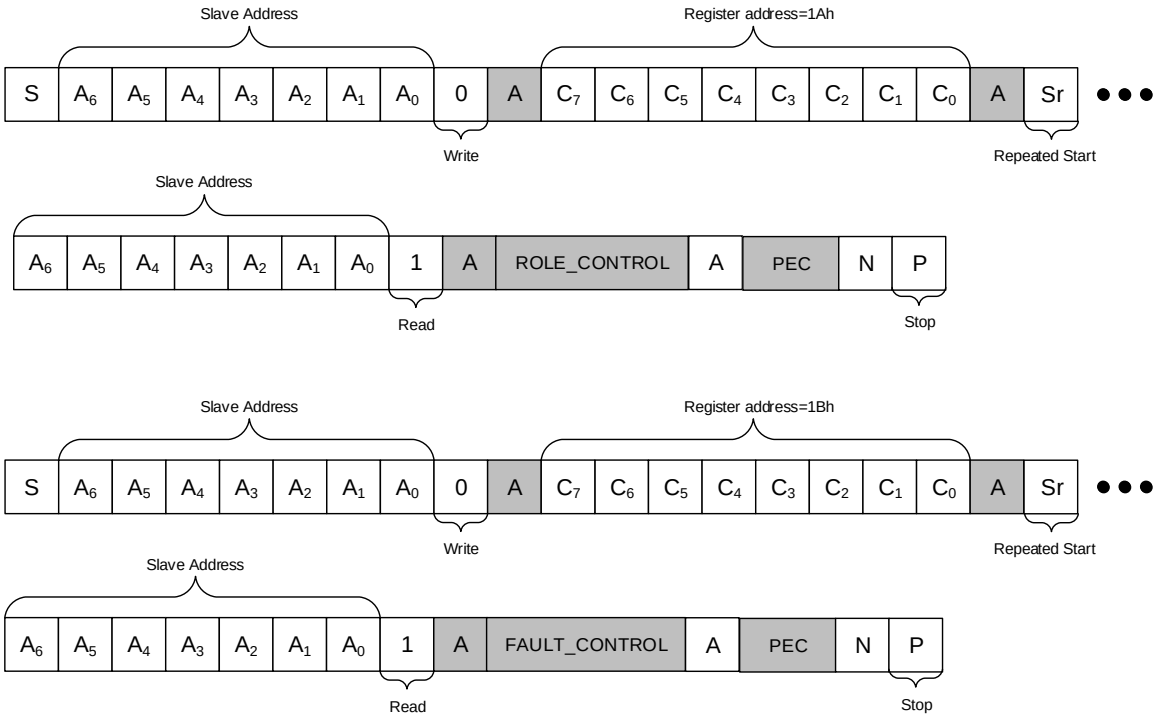


Figure 4-8. Reading Consecutive Single Byte Registers using SMBus PEC

4.3.9 Writing Two-Byte Registers using SMBus with PEC

Figure 4-9 illustrates how a 2 byte register can be written using SMBus with PEC.

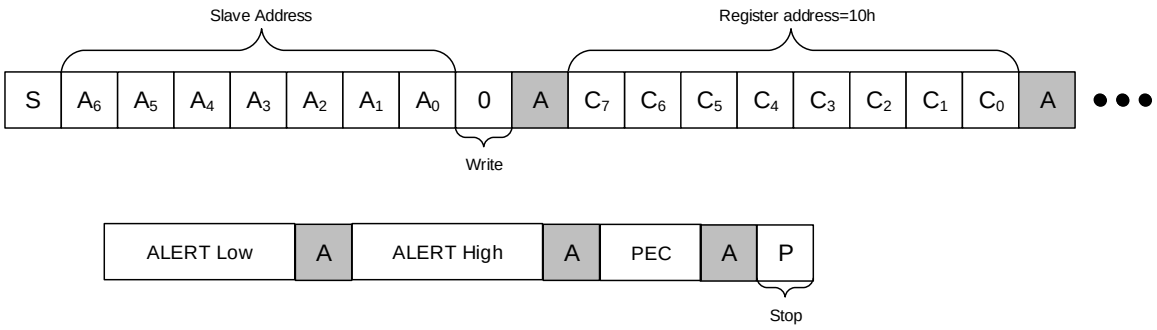


Figure 4-9. Writing a 2-Byte Register using SMBus PEC

4.3.10 Reading Two-Byte Registers using SMBus with PEC

Figure 4-10 illustrates how a 2 byte register can be read using SMBus with PEC.

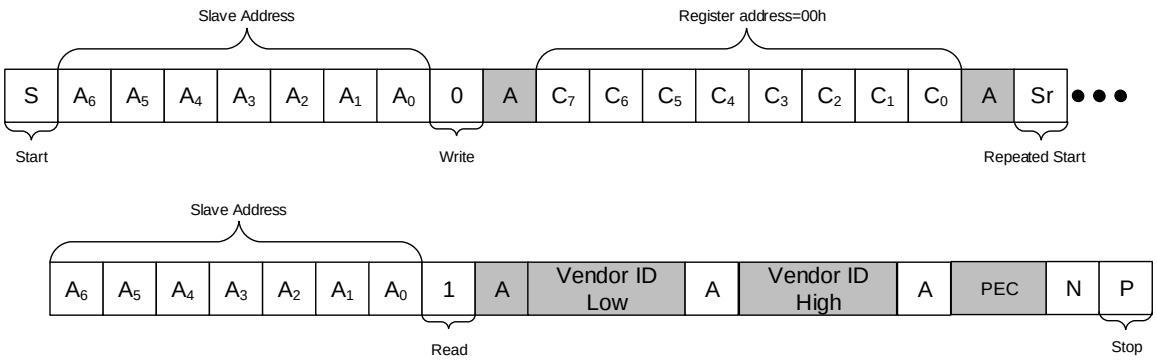


Figure 4-10. Reading a 2-Byte Register using SMBus PEC

4.3.11 Writing the TRANSMIT_BUFFER using SMBus with PEC

Figure 4-11 illustrates how the transmit buffer can be written using SMBus with PEC.

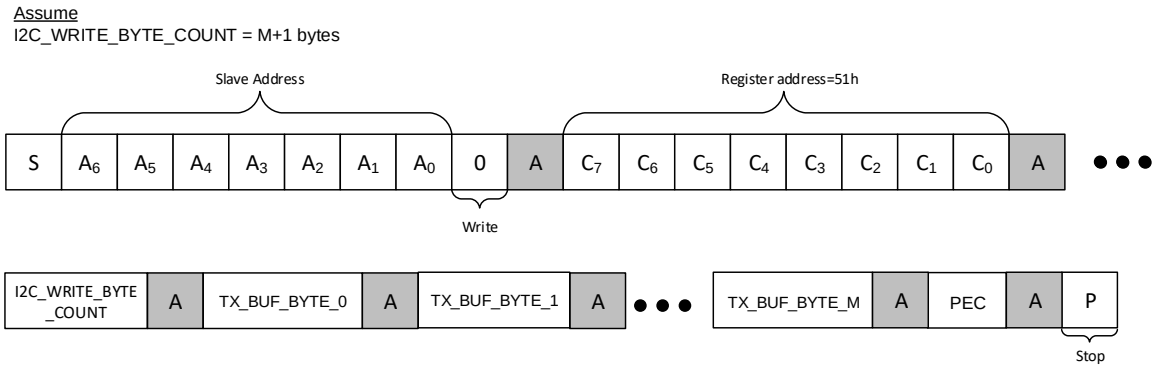


Figure 4-11. Writing the TRANSMIT_BUFFER using SMBus PEC

4.3.12 Reading the RECEIVE_BUFFER using SMBus with PEC

Figure 4-12 illustrates how the receive buffer can be read using SMBus with PEC.

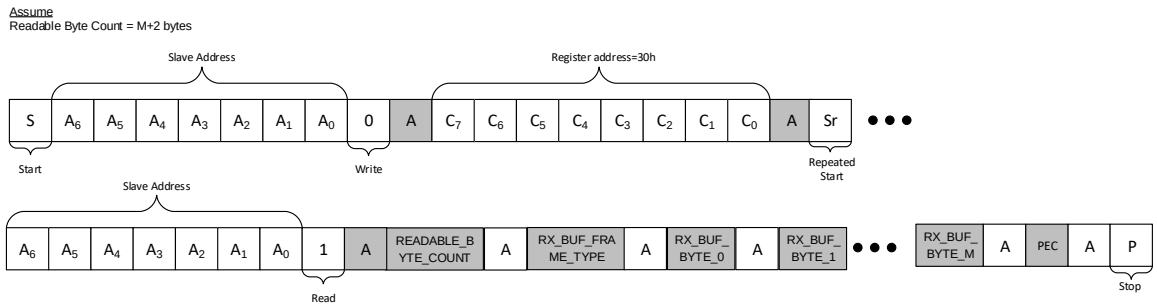


Figure 4-12. Reading the RECEIVE_BUFFER using SMBus PEC

4.4 Register Definition

This section defines the registers for the TCPC.

4.4.1 Identification Registers

4.4.1.1 VENDOR_ID (Normative)

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. 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 (Normative)

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. 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. 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 (Normative)

This register refers to ***USB Type-C*** Cable and Connector Specification Revision, ***USB Type-C*** represented by a unique 16-bit unsigned register. The format is packed binary coded decimal.
This specification revision 2.0 version 1.3 aligns with USB Type-C Release 2.1.

Table 4-5. USBTYPEC_REV Register Definition

Bit(s)	Name	Description
B15…8	Reserved	Set to 0
B7…0	bcdUSBTYPEC Release	Example: 0010 0001 – Release2.1*

*A TCPC vendor shall program this register in accordance with the ***USB Type-C*** specification the product supports. If the TCPC IC supports Type-C Release 2.1, this register shall be set to 2.1.

4.4.1.4 USBPD_REV_VER (Normative)

This register refers to ***USB PD*** specification Revision and Version, ***USB PD*** represented by a unique 16-bit unsigned integer. The format is packed binary coded decimal.
This specification revision 2.0 version 1.3 aligns with ***USB PD*** Revision 3.1 Version 1.0.

Table 4-6. USBPD_REV_VER Register Description

Bit(s)	Name	Description
B15..8	bcdUSBPD Revision	Example: 0011 0001 – Revision 3.1*
B7..0	bcdUSBPD Version	Example: 0001 0000 – Version 1.0

* A TCPC vendor shall program this register in accordance with the ***USB PD*** specification the product supports. If the TCPC IC supports ***USB PD*** Release 3.1 and Version 1.0, this register shall be set to 3.1 and 1.0 correspondingly.

4.4.1.5 USB-Port Controller Interface Specification Revision (Normative)

The USB-Port Controller 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. Only values that represent valid releases of the Universal Serial Bus Type-C ® Port Controller Interface Specification shall be used. The TCPC shall comply with all requirements of the revision and version in these fields.

Table 4-7. PD_INTERFACE_REV Register Description

Bit(s)	Name	Description
B15..8	bcd USB-PD Inter-Block Specification Revision	Example: 0010 0000 – Revision 2.0
B7..0	bcd USB-PD Inter-Block Specification Version	Example: 0001 0011 – Version 1.3 (See doc title for current version)

4.4.2 ALERT Register (Normative)

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. This register shall be initialized per Table 4-1 upon power on.
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. ALERT Register Definition

Bit(s)	Name	Description
B15	Vendor Defined Alert	0b: Cleared 1b: A vendor defined alert has been detected. Defined in the VENDOR_DEFINED registers. Refer to the vendor datasheet for details. This bit can be cleared, regardless of the current status of the alert source.
B14	Alert Extended	0b: Cleared 1b: An extended interrupt event has occurred. Read the ALERT_EXTENDED register.
B13	Extended Status	0b: Cleared, 1b: Extended Status changed
B12	Beginning SOP* Message Status	0b: Cleared, 1b: RECEIVE_BUFFER register changed. READABLE_BYTE_COUNT being set to 0 does not set this bit. Set if READABLE_BYTE_COUNT is greater than 133 to indicate an extended USB PD message with more than 128 data bytes has been received. Not set if READABLE_BYTE_COUNT is 133 or less.
B11	VBUS Sink Disconnect Detected	0b: Cleared 1b: The TCPC in Attached_Snk state has detected a Sink disconnect. This bit shall only be asserted when POWER_CONTROL.AutoDischargeDisconnect is set. The Sink TCPC asserts this bit either when POWER_STATUS.VbusPresent transitions from 1b to 0b (if DEVICE_CAPABILITIES_2.SinkDisconnectDetection=0b) or the TCPC detects VBUS falling below VBUS_SINK_DISCONNECT_THRESHOLD (if DEVICE_CAPABILITIES_2.SinkDisconnectDetection=1b).
B10	Rx Buffer Overflow	0b: TCPC Rx buffer is functioning properly 1b: TCPC Rx buffer has overflowed. Future GoodCRC shall not be sent. Writing 1 to this register acknowledges the overflow. The overflow is cleared by writing to ALERT.ReceiveSOP*MessageStatus

Bit(s)	Name	Description
B9	Fault	0b: No fault 1b: A fault has occurred. Read the FAULT_STATUS register
B8	VBUS Voltage Alarm Lo	0b: Cleared 1b: A low-voltage alarm has occurred
B7	VBUS Voltage Alarm Hi	0b: Cleared 1b: A high-voltage alarm has occurred
B6	Transmit SOP* Message Successful	0b: Cleared, 1b: Reset or SOP* message transmission successful. GoodCRC response received on SOP* message transmission. Transmit SOP* message buffer registers are empty.
B5	Transmit SOP* Message Discarded	0b: Cleared, 1b: Reset or SOP* message transmission not sent due to an incoming receive message. Transmit SOP* message buffer registers are empty.
B4	Transmit SOP* Message Failed	0b: Cleared, 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, 1b: Received Hard Reset message
B2	Received SOP* Message Status	0b: Cleared, 1b: RECEIVE_BUFFER register changed. READABLE_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 TCPC shall not assert this bit when CC_STATUS.Looking4Connection changes state if TCPC_CONTROL.EnableLooking4ConnectionAlert is set to 0.

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

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, FAULT_STATUS_MASK, EXTENDED_STATUS_MASK and ALERT_EXTENDED_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. An EXTENDED_STATUS change has to be unmasked in both the EXTENDED_STATUS_MASK and the ALERT.ExtendedStatusInterruptMask to enable the Alert# pin. An ALERT_EXTENDED change has to be unmasked in both the ALERT_EXTENDED _MASK and the ALERT.AlertExtendedInterruptMask to enable the Alert# pin.

4.4.3.1 ALERT_MASK (Normative)

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

Bit(s)	Name	Description
B15	Vendor Defined Alert	0b: Interrupt masked, 1b: Interrupt unmasked
B14	Alert Extended Interrupt Mask	0b: Interrupt masked, 1b: Interrupt unmasked
B13	Extended Status Interrupt Mask	0b: Interrupt masked, 1b: Interrupt unmasked
B12	Beginning SOP* Message Status	0b: Interrupt masked, 1b: Interrupt unmasked
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 (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

Bit(s)	Name	Description
B0	CC Status Interrupt Mask	0b: Interrupt masked, 1b: Interrupt unmasked

4.4.3.2 POWER_STATUS_MASK (Normative)

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. This register shall be initialized per Table 4-1 upon power on or Hard Reset. The assertion of the Alert# pin is prevented when the corresponding bit is set to zero by the TCPM.

Table 4-10. POWER_STATUS_MASK Register Definition

Bit(s)	Name	Description
B7	Debug Accessory Connected Status Mask	0b: Interrupt masked, 1b: Interrupt unmasked
B6	TCPC Initialization Status Mask	0b: Interrupt masked, 1b: Interrupt unmasked
B5	Sourcing Nondefault Voltage Status Interrupt Mask	0b: Interrupt masked, 1b: Interrupt unmasked
B4	Sourcing VBUS Status Interrupt Mask	0b: Interrupt masked, 1b: Interrupt unmasked
B3	VBUS 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 (Normative)

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 the Alert# pin is prevented when the corresponding bit is set to zero by the TCPM.
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 SIGNAL, VBUS External Over-Current Fault and the status reported in this register. An internal OCP fault shall be reported in this register if implemented. The action taken during OCP event is vendor defined.
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 SIGNAL, VBUS External Over-Voltage Fault and the status reported in this register. An internal OVP fault shall be reported in this register if implemented. The action taken during OVP event is vendor defined.

Table 4-11. FAULT_STATUS_MASK Register Definition

Bit(s)	Name	Description
B7	AllRegistersResetToDefault	0b: Interrupt masked, 1b: Interrupt unmasked The condition that generates a FAULT_STATUS.AllRegistersResetToDefault Interrupt also resets this bit; therefore, writing a 0b to this bit will not mask FAULT_STATUS.AllRegistersResetToDefault Interrupt.
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 VBUS Over Current Protection Fault Interrupt Status Mask	0b: Interrupt masked, 1b: Interrupt unmasked
B2	Internal or External OVP 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

4.4.3.4 EXTENDED_STATUS_MASK (Normative)

This is an event interrupt mask. It is masked and unmasked by the TCPM. The EXTENDED_STATUS_MASK register is cleared by the TCPM. This register shall be initialized per Table 4-1 upon power on or Hard Reset.
The assertion of the Alert# pin is prevented when the corresponding bit is set to zero by the TCPM.

Table 4-12. EXTENDED_STATUS_MASK Register Definition

Bit(s)	Name	Description
B7…1	Reserved	Shall be set to zero by sender and ignored by receiver
B0	vSafe0V Status Mask	0b: Interrupt masked, 1b: Interrupt unmasked

4.4.3.5 ALERT_EXTENDED_MASK (Normative)

This is an event interrupt mask. It is masked and unmasked by the TCPM. The ALERT_EXTENDED_MASK register is cleared by the TCPM. This register shall be initialized per Table 4-1 upon power on or Hard Reset.
The assertion of the Alert# pin is prevented when the corresponding bit is set to zero by the TCPM.

Table 4-13. ALERT_EXTENDED_MASK Register Definition

Bit(s)	Name	Description
B7…3	Reserved	Shall be set to zero by sender and ignored by receiver
B2	Timer Expired	0b: Interrupt masked, 1b: Interrupt unmasked

Bit(s)	Name	Description
B1	Source Fast Role Swap Mask	0b: Interrupt masked, 1b: Interrupt unmasked
B0	Sink Fast Role Swap Mask	0b: Interrupt masked, 1b: Interrupt unmasked

4.4.4 CONFIGURE STANDARD OUTPUT (Optional Normative)

This register is required if any Standard Outputs are declared in the STANDARD_OUTPUT_CAPABILITIES register (Section 4.4.9.4). 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 SIGNALs defined in Table 4-48. The Standard Outputs shall reset to open-drain per Table 4-1.

Table 4-14. CONFIG_STANDARD_OUTPUT Register Definition

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

4.4.5 Control and Configuration Registers

4.4.5.1 TCPC_CONTROL (Normative)

After the TCPC has set the power on reset default values per Table 4-1, this register is set and cleared only by the TCPM. 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 the TCPC clock stretching on the I2C bus. Allowing clock stretching may result in lower power from the TCPC, but can 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 regardless of the clock stretching setting chosen by the TCPM, unless the TCPM has put it to sleep using COMMAND.I2CIdle, or the TCPM writes to a register/bit that is not implemented/reserved.

Table 4-15. TCPC_CONTROL Register Definition

Bit(s)	Name	Description
B7	Enable SMBus PEC	0b: SMBus PEC is disabled (default) 1b: SMBus PEC is enabled Enables SMBus PEC according to Section 4.1.
B6	Enable Looking4Connection Alert	0b: Disable ALERT.CcStatus assertion when CC_STATUS.Looking4Connection changes (default) 1b: Enable ALERT.CcStatus assertion when CC_STATUS.Looking4Connection changes
B5	Enable Watchdog Timer	0b: Watchdog Monitoring is disabled (default) 1b: Watchdog Monitoring is enabled Enables Watchdog Timer Monitoring according to Section 3.8 Required if DEVICE_CAPABILITIES_2.WatchDogTimer = 1b
B4	Debug Accessory Control	0b: Controlled by TCPC (power on default) 1b: Controlled by TCPM. The TCPM writes 1b to this register to take over control of asserting the DebugAccessoryConnected#. See Table 4-14. Required (register is required but output is not required)
B3..2	I2C Clock Stretching Control	Clock Stretching Control 00b: Disable clock stretching. TCPC shall not perform any clock stretching during I2C transfers. 01b: Reserved 10b: Enable clock stretching. TCPC is allowed limited clock stretching during each I2C Transfer. 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. The TCPC datasheet should contain details of the power consequences of clock stretching as well as the max duration of clock stretching per I2C transaction. The TCPC shall limit total clock stretching as detailed in Section 4.10. This feature is optional. The TCPC is allowed to ignore updates to these bits if it has not implemented clock stretching. The power on default disable clock stretching.

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 disconnect is detected. 0: Normal Operation. Incoming messages enabled by RECEIVE_DETECT passed to TCPM via Alert. 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. 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 message delivery is enabled. 1b: When VCONN is enabled, apply it to the CC1 pin. Monitor the CC2 pin for BMC communications if PD message delivery is enabled. Required

4.4.5.2 ROLE_CONTROL (Normative)

After the TCPC has set the power on reset default values per Table 4-17, this register is set and cleared only by the TCPM. The TCPM writes to this register to configure the CC pull up (Rp) or pull down (Rd) resistors.
The TCPM shall 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. When controlling Rp/Rd directly, the TCPM writes to B3..0 (CC1/CC2) each time it wishes to change the CC1/CC2 values. This control is used for TCPM-TCPC implementing Source or Sink only as well as when a connection has been detected via DRP toggling but the TCPM wishes to attempt Try.SRC or Try.SNK (as defined in ***USB Type-C***).
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, the TCPM shall write B6 (DRP) =1b and write 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 while POWER_CONTROL.AutoDischargeDisconnect = 0b, as in Figure 4-19. It is recommended the TCPM write ROLE_CONTROL.DRP=0 before writing to POWER_CONTROL.AutoDischargeDisconnect and starting the DRP toggling using COMMAND.Look4Connection as shown in Figure 4-24, Figure 4-25 and Figure 4-26.
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.
When CC1 and CC2 are set to Open and DRP = 0b, the TCPC may power down the PHY and CC Status comparators.
When the TCPM has set ROLE_CONTROL.CC1 = ROLE_CONTROL.CC2 = 11b (Open), the Sink TCPC is recommended to remove terminations from the CC pins as long as the VBUS is present (i.e. above vSafe5V). This allows a bus-powered Sink TCPC to remove terminations from CC pins in ***USB Type-C*** ErrorRecovery state when the VBUS is present.

Table 4-16. 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 1b: DRP 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 4-19 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. The TCPC shall stay in Potential_Connect_as_Src or Potential_Connect_as_Sink until directed otherwise.
B5..4	Rp Value	00b: Rp default 01b: Rp 1.5A 10b: Rp 3.0A 11b: Reserved
B3..2	CC2	00b: Ra 01b: Rp (Use Rp definition in B5..4) 10b: Rd 11b: Open (Disconnect or don't care)
B1..0	CC1	00b: Ra 01b: Rp (Use Rp definition in B5..4) 10b: Rd 11b: Open (Disconnect or don't care)

Table 4-17 defines the power on default for ROLE_CONTROL and MESSAGE_HEADER_INFO.

Table 4-17. Power On Default Conditions

DEVICE_CAPABILITIES. RolesSupported	ROLE_CONTROL (Default)	MESSAGE_HEADER_INFO (Default)
Source or Sink (000b)	0Ah	04h
Source only (001b)	05h – Not dead battery N/A – Dead battery	0Dh
Sink only (010b)	0Ah	04h
Sink with Accessory (011b)	0Ah	04h
DRP (100b)	0Ah – Dead battery 4Ah – Not dead battery and DebugAccessoryIndicator supported 0Fh – Not dead battery and DebugAccessoryIndicator not supported	04h
Source, Sink, DRP (101b and 110b) Applies to SOP Devices	0Ah – Source, Sink, or DRP dead battery 4Ah – Not dead battery and DebugAccessoryIndicator supported 0Fh – Not dead battery and DebugAccessoryIndicator not supported	04h

4.4.5.3 FAULT_CONTROL (Normative)

After the TCPC has set the power on reset default values per Table 4-1, this register is set and cleared only by the TCPM. The TCPM writes to FAULT_CONTROL to enable/disable the FAULT circuitry.

Table 4-18. 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 (Source or Sink)	0b: Allow STANDARD INPUT SIGNAL Force Off Vbus control (default) 1b: Block STANDARD INPUT SIGNAL Force Off Vbus control This enables or disables the STANDARD INPUT SIGNAL Force Off Vbus (Section 4.5.1) functionality for debug purposes. Required if STANDARD_INPUT_CAPABILITES.ForceOffVBUS = 1b.
B3	VBUS Discharge Fault Detection Timer	0b: VBUS Discharge Fault Detection Timer enabled 1b: VBUS Discharge Fault Detection Timer disabled This enables the timers for both FAULT_STATUTS.AutoDischargeFailed and FAULT_STATUS.ForceDischargeFailed Required
B2	Internal or External OCP VBUS Over Current Protection Fault	0b: Internal and External OCP circuit enabled 1b: Internal and External OCP circuit disabled Required if DEVICE_CAPABILIITES_1.VBUSOCPReporting = 1b
B1	Internal or External OVP VBUS Over Voltage Protection Fault	0b: Internal and External OVP circuit enabled 1b: Internal and External OVP circuit disabled Required if DEVICE_CAPABILIITES_1.VBUSOVPReporting = 1b
B0	VCONN Over Current Fault	0b: Fault detection circuit enabled 1b: Fault detection circuit disabled Required if DEVICE_CAPABILITIES_2.VCONNOvercurrentFaultCapable = 1b

4.4.5.4 POWER_CONTROL (Normative)

After the TCPC has set the power on reset default values per Table 4-1, this register is set and cleared by the TCPM.

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

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

The TCPM shall take the following steps to request sourcing VCONN over one of the CC pins:

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

The TCPC shall source VCONN as TCPM requested irrespective of the status of VBUS and CC1, CC2 wires. The TCPC shall not autonomously disable VCONN sourcing when VBUS is removed, or CC1, CC2 status has changed. The TCPM shall set POWER_CONTROL.EnableVCONN=0b to request disabling VCONN sourcing. When a Source Only TCPC is requested to disable VCONN sourcing (by setting POWER_CONTROL.EnableVCONN=0b), the Source Only TCPC shall apply Rdch termination on the CC pin that was providing VCONN until the CC pin is discharged below vVCONNDischarge level as specified in the ***USB Type-C***. A DRP TCPC may not provide the capability of applying Rdch termination when disabling VCONN sourcing is requested.

Table 4-19. POWER_CONTROL Register Definition

Bit(s)	Name	Description
B7	Fast Role Swap Enable	0b: Disable Fast Role Swap function 1b: Enable Fast Role Swap function Sink TCPC shall support this bit if DEVICE_CAPABILITIES_2.SinkFRSwap = 1b. Source TCPC shall support this bit if DEVICE_CAPABILITIES_2.SourceFRSwap = 1b. The detailed functional requirements for Source and Sink TCPCs supporting Fast Role Swap are provided in Section 4.4.5.4.6.
B6	VBUS_VOLTAGE Monitor	0b: VBUS_VOLTAGE Monitoring is enabled 1b: VBUS_VOLTAGE Monitoring is disabled (default) Controls only VBUS_VOLTAGE Monitoring. VBUS_VOLTAGE shall report all zeroes if disabled. Required if DEVICE_CAPABILITIES_1.VBUSMeasurement&AlarmCapable = 1b
B5	Disable Voltage Alarms	0b: Voltage Alarms Power status reporting is enabled 1b: Voltage Alarms Power status reporting is disabled (default) Controls VBUS_VOLTAGE_ALARM_HI_CFG and VBUS_VOLTAGE_ALARM_LO_CFG. Required if DEVICE_CAPABILITIES_1.VBUSMeasurement&AlarmCapable = 1b
B4	Auto Discharge Disconnect	0b: The TCPC shall not automatically discharge VBUS based on VBUS voltage (default) 1b: The TCPC shall automatically discharge Refer to Sections 4.4.5.4.1 and 4.4.5.4.2 for more details. Setting this bit in a Source TCPC triggers the following actions upon a disconnect detection: 1. Disable sourcing power over VBUS 2. VBUS discharge Sourcing power over VBUS shall be disabled before or at the same time as starting VBUS discharge. Setting this bit in a Sink TCPC triggers the following action upon a disconnect detection: 1. VBUS discharge The TCPC shall automatically disable discharge (without clearing this bit) once the voltage on VBUS is below vSafe0V (max). Disconnect detection is defined in Section 4.4.5.4.2. TCPC shall not re-apply discharge circuit if VBUS rises above vSafe0V. Required
B3	Enable Bleed Discharge	0b: Disable bleed discharge of VBUS (default) 1b: Enable bleed discharge of VBUS The bleed discharge is a low current load on VBUS where the recommended loading is either 10kΩ or 2mA. Refer to Section 4.4.5.4.5 Required if DEVICE_CAPABILITIES_1.BleedDischarge = 1b
B2	Force Discharge	0b: Disable forced discharge (default) 1b: Enable forced discharge of VBUS. Refer to Section 4.4.5.4.3 Required if DEVICE_CAPABILITIES_1.ForceDischarge = 1b

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

Table 4-20. Discharge Timing Parameters

Name	Description	Min	Max	Units
tDisconnectDetect	Time from disconnect to detection of a disconnect		6	ms

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

When in Attached_Src state in Figure 4-19 and Figure 4-20, the TCPC shall fully discharge VBUS to vSafe5V (max) within tSafe5V and then to vSafe0V within tSafe0V when a disconnect occurs. A TCPC in the Attached_Src state (as shown in Figure 4-19 and Figure 4-20) shall detect a disconnect if the CC State of 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)

A TCPC in Attached_Snk state in Figure 4-19 and Figure 4-20 shall use either the POWER_STATUS.VbusPresent transition from 1b to 0b, or VBUS falling below VBUS_SINK_DISCONNECT_THRESHOLD (Table 4-42) as a Sink disconnect indicator. The mechanism used shall be defined in DEVICE_CAPABILITIES_2.SinkDisconnectDetection. The Sink TCPC shall detect a cable removal within tDisconnectDetect (Table 4-20) 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 read EXTENDED_STATUS.vSafe0V = 1b.
When in Sink mode and POWER_CONTROL.AutoDischargeDisconnect=1b, the TCPC shall fully discharge VBUS to vSafe5V (max) within tSafe5V and then to vSafe0V (max) within tSafe0V of the removal of the cable. The TCPC shall not re-apply the discharge circuit if VBUS rises above vSafe0V (max). Stop discharge is an edge-triggered event.
When the Source is removed, the system load and optionally the bleed discharge circuit will discharge the Sink bulk capacitance cSnkBulkPd. The time required to discharge cSnkBulkPd to below the disconnect detection threshold is tSinkDischargeBleed and it is dependent upon the strength of the system load and optionally the bleed discharge. The time required to discharge the Sink bulk capacitance to vSafe5V is dependent upon the strength of the full discharge, the system load and optionally the bleed discharge. The total time tSinkDischargeBleed + tSinkDischargeFull 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 10kΩ,
- the Sink bulk capacitance cSnkBulkPd is 82µF,
- there is no system load,
- the initial voltage is 21.5V (20V + 5% + vSrcValid(max)),

- the minimum valid VBUS voltage seen by Sink when negotiated through USB PD, vSinkPD_min = 17.75V = 0.95*20V – 750mV + vSrcValid(min)
- and VBUS_SINK_DISCONNECT_THRESHOLD = 15.975V (90% of vSinkPD_min)

then tSinkDischargeBleed = 199ms + tDisconnectDetect(max) = 205ms. Once the VBUS_SINK_DISCONNECT_THRESHOLD has been reached, the Sink connects its full discharge resistance of 400Ω. The Sink then discharges to vSafe5V in 36.5ms and vSafe0V in 99.7ms. The total time to reach vSafe5V is then tSinkDischargeBleed + tSinkDischargeFull = 242ms. The total time to reach vSafe0V is then 305ms. Both tSafe5V and tSafe0V can be met. It is assumed the Sink bulk capacitance (82μF) does not change during the voltage transitions.

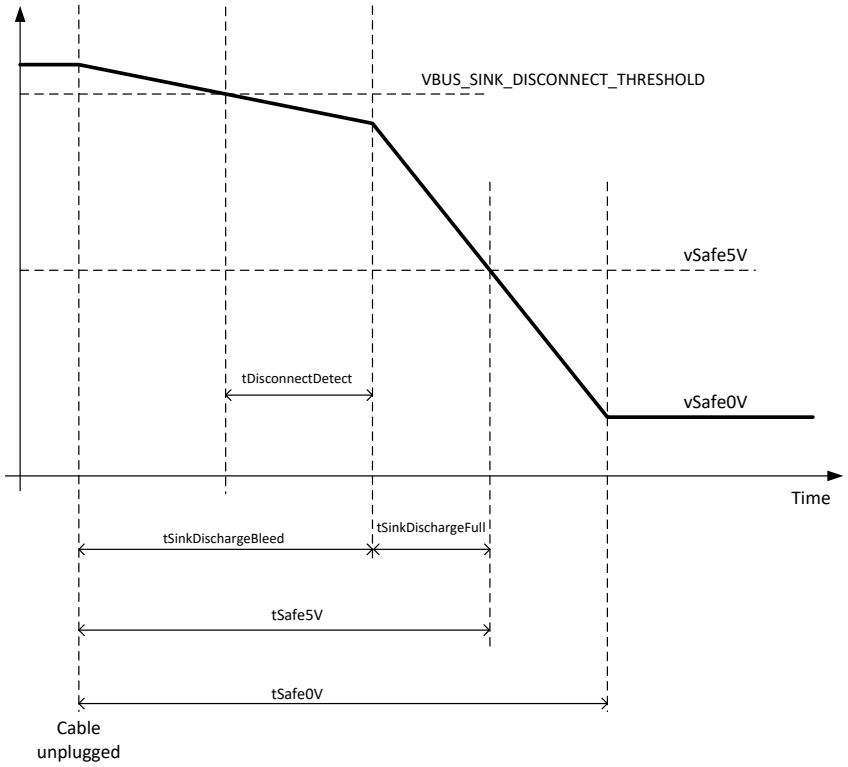


Figure 4-13. Automatic VBUS Sink Discharge by the TCPC after a Disconnect

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 Force Discharge circuit (without clearing POWER_CONTROL.ForceDischarge bit) once the voltage on VBUS is below the value indicated by VBUS_STOP_DISCHARGE_THRESHOLD (if DEVICE_CAPABILITIES_2. StopDischargeThreshold=1b) or vSafe0V (if DEVICE_CAPABILITIES_2. StopDischargeThreshold=0b).
A Source TCPC transitioning from a higher to lower voltage shall remove the Force Discharge circuit in time to meet the ***USB PD*** vSrcValid requirement. The TCPC shall not re-apply the Force Discharge circuit if VBUS rises above VBUS_STOP_DISCHARGE_THRESHOLD or vSafe0V. Stop discharge is an edge-triggered event.
The TCPC shall discharge VBUS to vSafe0V after a power on reset before applying the Rp.

4.4.5.4.4 Disconnect Detection by the Sink TCPC during a Connection (Optional Normative)

Upon reception of or prior to transmitting a PR_Swap message, the TCPM acting as a Sink shall disable the Sink disconnect detection to retain PD message delivery when Power Role Swap happens. Similarly, upon setting POWER_CONTORL.FastRoleSwapEnable (i.e deciding

to support Fast Role Swap), the TCPM acting as a Sink shall disable the Sink disconnect detection to retain PD message delivery when Fast Role Swap happens.
The steps of disabling Sink disconnect detection shall be as follow:

1. TCPM sets POWER_CONTROL.AutoDischargeDisconnect = 0b
2. TCPM sets RECEIVE_DETECT.MessageDisableDisconnect = 0b if this bit was previously set to 1b

4.4.5.4.5 Bleed Discharge (Optional Normative)

The bleed discharge circuit is enabled and disabled by the TCPM. The bleed discharge is a low current load on the VBUS.

4.4.5.4.6 Fast Role Swap (Optional Normative)

Setting the POWER_CONTROL.FastRoleSwapEnable bit in the Sink TCPC triggers the following actions when receiving Fast Role Swap signal. Required if DEVICE_CAPABILITIES_2.SinkFRSwap = 1b.

1. Set ALERT_EXTENDED.SinkFRSwap = 1b.
2. Disable the Sink path (equivalent to COMMAND.DisableSinkVbus)
3. When VBUS falls below vSafe5V(max), execute sourcing vSafe5V over VBUS (equivalent to COMMAND.SourceVbusDefaultVoltage). Note that the initial Sink shall meet the *tScrFRSwap* timing requirement as specified in the ***USB PD***.

Setting POWER_CONTROL.FastRoleSwapEnable bit in Source TCPC enables one of the following actions. Required if DEVICE_CAPABILITIES_2.SourceFRSwap = 1b.

A. The TCPC shall send a Fast Role Swap signal within tTCPCSendFRSwap after receiving COMMAND.SendFRSwapSignal.

or,

B. If CONFIG_EXTENDED1.StandardInputSourceFRSwap is set, the TCPC shall send a Fast Role Swap signal within tTCPCSendFRSwap when STANDARD INPUT SIGNAL Source FR Swap is set low. After the Fast Role Swap signal is sent, the TCPC shall set ALERT_EXTENDED.SourceFRSwap to 1. Required if STANDARD_INPUT_CAPABILITIES.SourceFRSwap is either set to 01b or 10b.

Figure 4-14 shows the interactions between the Source and Sink TCPCs during a Fast Role Swap operation. It shows a flow where the initial Source TCPC starts the process “send Fast Role Swap signal” after receiving COMMAND.SendFRSwapSignal. Figure 4-15 indicates a flow where the Fast Role Swap signal transmission is triggered by STANDARD INPUT SIGNAL Source FR Swap being set low.
USB PD message passing after the FR_Swap Message is sent, is not shown (in Figure 4-14 and Figure 4-15) because the Accept Message (from the initial Source in response to FR_Swap Message) may be sent before or after the VBUS drops to vSafe5V. It is also possible that the FR_Swap Message is sent after the VBUS drops to vSafe5V. That is, the initial Sink may execute sourcing vSafe5V over VBUS before FR_Swap Message is sent.
It is assumed the initial Source implements a bidirectional power path hence the power path switchover time is not shown in Figure 4-14 and Figure 4-15. When there are separate Sink and Source power paths, the initial Source should disable the Source path without enabling the VBUS discharge circuity when sending the Fast Role Swap signal, and then the initial Source should start to enable the Sink path.

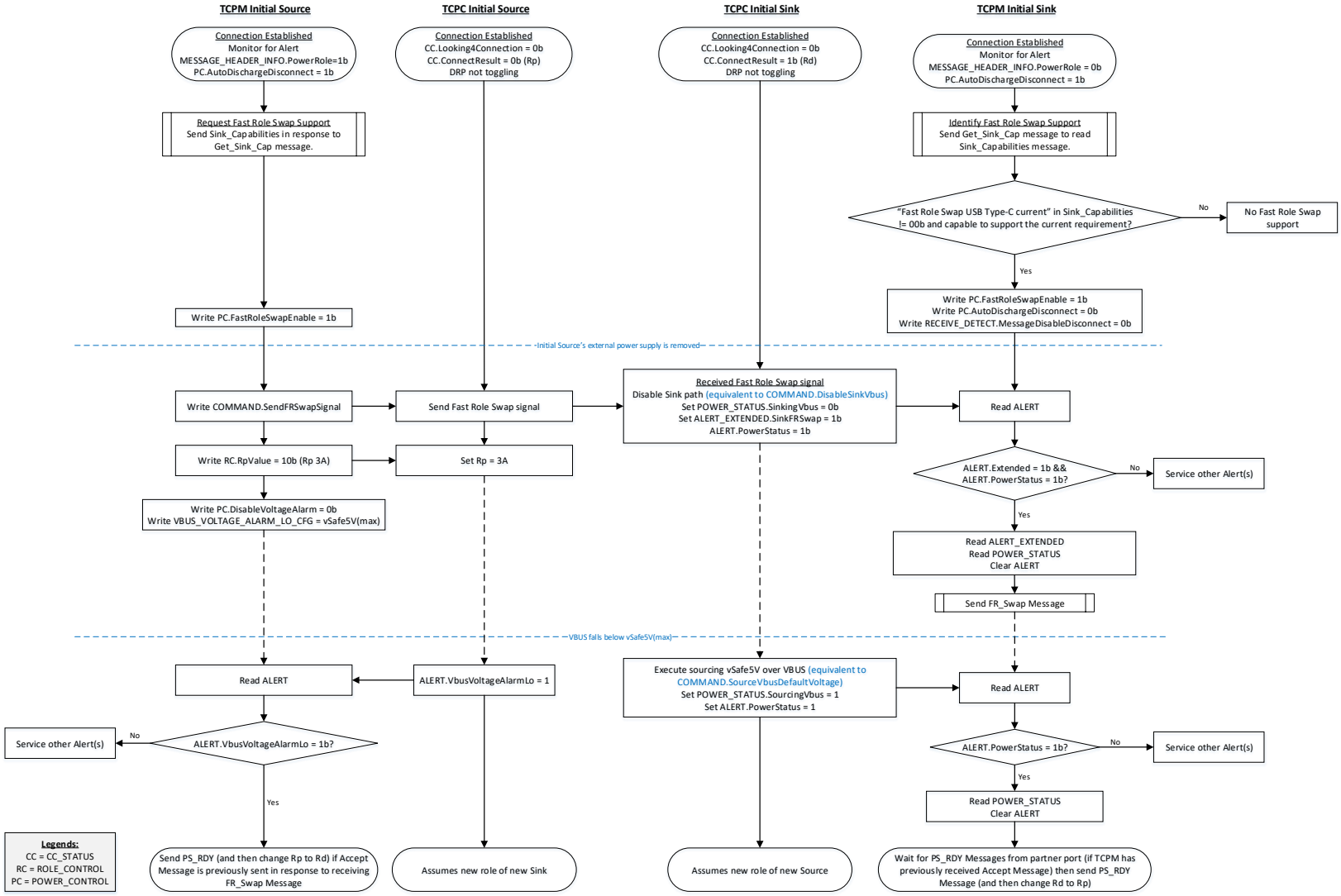


Figure 4-14. COMMAND.SendFRSwapSignal triggered Fast Role Swap operation

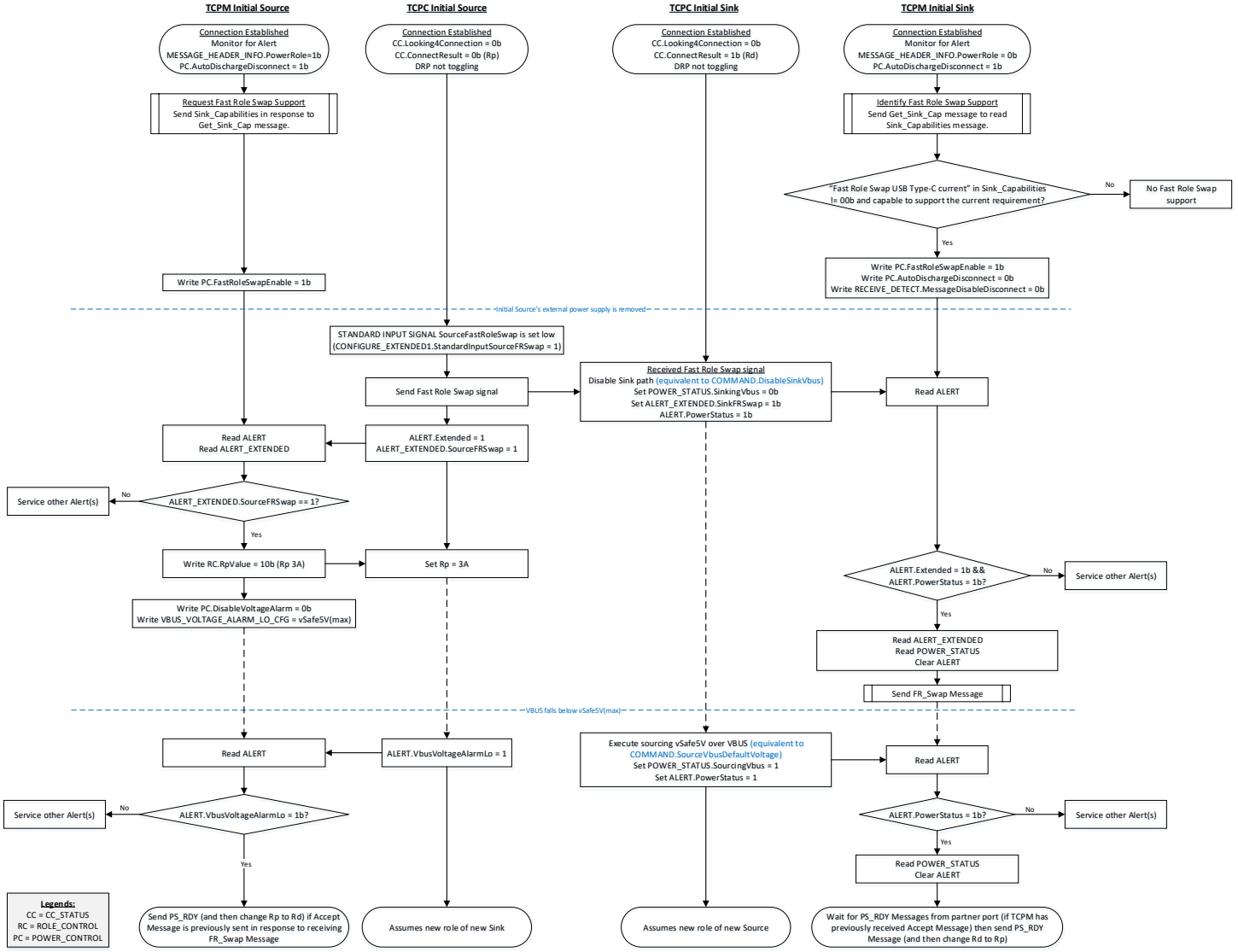


Figure 4-15. STANDARD INPUT SIGNAL Source FR Swap triggered Fast Role Swap operation

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 (Normative)

This register is set and cleared by the TCPC. The TCPC shall update the CC_STATUS register within tSetReg (Table 4-49) 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 Figure 4-24 and Section 4.4.8.
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***.
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) (Table 4-49) 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 4-21. Debounce requirements

	Min	Max	Units
tTCPCfilter	250	500	µs

Table 4-22. 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. 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 1b: the TCPC is presenting Rd

B3…2	CC2 State	If (ROLE_CONTROL.CC2=Rp) or (ConnectResult=0)
		00b: SRC.Open (open, Rp)
		01b: SRC.Ra (below maximum vRa)
		10b: SRC.Rd (within the vRd range)
		11b: reserved
		
		If (ROLE_CONTROL.CC2=Rd) or (ConnectResult=1)
		00b: SNK.Open (below maximum vRa)
		01b: SNK.Default (above minimum vRd-Connect)
		10b: SNK.Power1.5 (above minimum vRd-Connect), detects Rp 1.5A
		11b: SNK.Power3.0 (above minimum vRd-Connect), detects Rp 3.0A
		
		If ROLE_CONTROL.CC2=Ra, this field is set to 00b
		If ROLE_CONTROL.CC2=Open, this field is set to 00b
		
		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)
		00b: SRC.Open (open, Rp)
		01b: SRC.Ra (below maximum vRa)
		10b: SRC.Rd (within the vRd range)
		11b: reserved
		
		If (ROLE_CONTROL.CC1 = Rd) or ConnectResult=1)
		00b: SNK.Open (below maximum vRa)
		01b: SNK.Default (above minimum vRd-Connect)
		10b: SNK.Power1.5 (above minimum vRd-Connect), detects Rp-1.5A
		11b: SNK.Power3.0 (above minimum vRd-Connect), detects Rp-3.0A
		
		If ROLE_CONTROL.CC1=Ra, this field is set to 00b
		If ROLE_CONTROL.CC1=Open, this field is set to 00b
		
		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.

4.4.6.2 POWER_STATUS (Normative)

This register is set and cleared by the TCPC. The TCPM reads this register upon detecting an Alert# and reading the ALERT.PowerStatus bit 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*** Contract) shall detect VBUS presence and removal by reading the VBUSPresent bit.
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 4-23. POWER_STATUS Register Definition

Bit(s)	Name	Description
B7	Debug Accessory Connected	0b: No Debug Accessory connected (default) 1b: Debug Accessory connected Reflects the state of the DebugAccessoryConnected# output if supported Required (register is required but output is not required)
B6	TCPC Initialization Status	0b: The TCPC has completed initialization and all registers are valid 1b: The TCPC is still performing internal initialization and the only registers that are guaranteed to return the correct values are 00h...0Fh Required
B5	Sourcing Nondefault Voltage	0b: vSafe5V 1b: Nondefault VBUS voltage This bit does not control the power path, it just provides a monitor of the status. This bit is asserted as long as the TCPC is sourcing nondefault voltage over VBUS (i.e. not vSafe5V) as a response to TCPM writing to COMMAND.SourceNondefaultVbusVolage. Required if nondefault VBUS voltage can be sourced. This bit is not valid if POWER_STATUS.SourcingVbus = 0b.
B4	Sourcing VBUS	0b: Sourcing VBUS is disabled 1b: Sourcing VBUS is enabled This bit does not control the path, just provides a monitor of the status. Required
B3	VBUS Detection Enabled	0b: VBUS Detection Disabled 1b: VBUS Detection Enabled (default) Indicates whether the TCPC is monitoring for VBUS Present and vSafe0V level, or the detection circuits have been powered off Required
B2	VBUS Present	0b: VBUS Disconnected 1b: VBUS Connected 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. This bit is not valid when POWER_STATUS.VbusDetectionEnabled = 0b. Required

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

4.4.6.3 FAULT_STATUS (Normative)

This register is set by TCPC and cleared by TCPM. 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 4-24. FAULT_STATUS Register Definition

Bit(s)	Name	Description
B7	AllRegistersResetToDefault	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.
B6	Force Off VBUS (Source or Sink)	0b: No Fault Detected, no action (default or not supported) 1b: VBUS Source/Sink has been forced off due to external fault The TCPC has disconnected VBUS due to STANDARD_INPUT.ForceOffVbus. Required if STANDARD_INPUT_CAPABILITIES.ForceOffVbus = 1b
B5	Auto Discharge Failed	0b: No discharge failure 1b: Discharge commanded by the TCPM failed If POWER_CONTROL.AutoDischargeDisconnect is set, the TCPC shall report discharge fails if VBUS is not below vSafe0V within tSafe0V. Required
B4	Force Discharge Failed	0b: No discharge failure 1b: Discharge commanded by the TCPM failed If DEVICE_CAPABILITIES_2.StopDischargeThreshold=0b and POWER_CONTROL.ForceDischarge is set, the TCPC shall report a discharge fails if VBUS is not below vSafe0V within tSafe0V. Required if DEVICE_CAPABILITIES_1.ForceDischarge =1b

Bit(s)	Name	Description
B3	Internal or External OCP VBUS Over Current Protection Fault	0b: Not in an over-current protection state 1b: Over-current fault latched Required if DEVICE_CAPABILIITES_1.VBUSOCPReporting = 1b
B2	Internal or External OVP VBUS Over Voltage Protection Fault	0b: Not in an over-voltage protection state 1b: Over-voltage fault latched. Required if DEVICE_CAPABILIITES_1.VBUSOVPReporting = 1b
B1	VCONN Over Current Fault	0b: No Fault detected 1b: Over current VCONN fault latched Required if DEVICE_CAPABILITIES_2.VCONNOvercurrentFaultCapable = 1b
B0	I2C Interface Error	0b: No Error 1b: I2C error has occurred. The following conditions shall cause TCPC asserting this bit: • TCPM writes to the TRANSMIT register when the TRANSMIT_BUFFER is empty • The watchdog timer has expired • TCPM writes an invalid COMMAND • TCPM writes to the TRANSMIT_BUFFER when TCPC is transmitting the Fast Role Swap signal as triggered by the STANDARD INPUT SIGNAL Source Fast Role Swap • TCPM writes to CONFIG_EXTENDED1.FRSwapBidirectionalPin and STANDARD_INPUT_CAPABILITIES.SourceFastRoleSwap is not 10b The TCPC may assert this bit when TCPM writes a non-zero value to a reserved bit in a register. Required

4.4.6.4 EXTENDED_STATUS (Normative)

This register is set and cleared by the TCPC. The TCPM reads this register upon detecting an Alert# and reading the ALERT.ExtendedStatus bit set to 1.

Table 4-25. EXTENDED_STATUS Register Definition

Bit(s)	Name	Description
B7..1	Reserved	Shall be set to zero by sender and ignored by receiver
B0	vSafe0V	0b: VBUS is not at vSafe0V 1b: VBUS is at vSafe0V The TCPC shall report VBUS is at vSafe0V when TCPC detects VBUS is below 0.8V. This bit is not valid when POWER_STATUS.VbusDetectionEnabled = 0b. Required

4.4.7 ALERT_EXTENDED (Normative)

This register is set by TCPC and cleared by TCPM. The TCPM reads this register upon detecting an Alert# and reading the ALERT.AlertExtended bit set to 1.
The TCPM clears an ALERT_EXTENDED bit by writing a logical 1 to the respective bit position and then writing a logical 1 to the ALERT.AlertExtended bit after all bits in ALERT_EXTENDED have been cleared. The TCPM can clear any number of ALERT_EXTENDED 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_EXTENDED bit has no effect and therefore does not cause those ALERT_EXTENDED bits to be set or cleared.

Table 4-26. ALERT_EXTENDED Register Description

Bit(s)	Name	Description
B7…3	Reserved	Shall be set to zero by sender and ignored by receiver
B2	Timer Expired	0b: Generic timer is not expired 1b: Generic timer has expired
B1	Source Fast Role Swap	0b: No Fast Role Swap signal sent 1b: Fast Role Swap signal sent due to STANDARD INPUT Source Fast Role Swap is set low
B0	Sink Fast Role Swap	0b: No Fast Role Swap signal received 1b: Fast Role Swap signal received

4.4.8 COMMAND (Normative)

The COMMAND register is issued and written by the TCPM. The COMMAND register 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. 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. If POWER_CONTROL.AutoDischargeDisconnect is not set to 0b, 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 (as defined in ***USB Type-C***). 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 disabled by writing to the ROLE_CONTROL register, but its behavior is not affected by the COMMAND.I2Cidle.
While there is a valid Source-to-Sink connection, the TCPM acting as a Sink shall write COMMAND.DisableSinkVbus (if DEVICE_CAPABILITIES_1.SinkVbus = 1b) to remove the Sink connection upon reception of or prior to transmitting a Power Role Swap or Hard Reset.
The TCPM shall issue COMMAND.SourceVbusNondefaultVoltage to enable the TCPC to transition the VBUS source to a nondefault voltage level. VBUS_NONDEFAULT_TARGET is an optional register declared in DEVICE_CAPABILITIES_1 register. The target voltage for COMMAND.SourceVbusNondefaultVoltage can be set in the VBUS_NONDEFAULT_TARGET register. Alternatively, the target voltage level for COMMAND.SourceVbusNondefaultVoltage is set in a vendor defined manner. The steps of transitioning to source a nondefault voltage over VBUS using NONDEFAULT_VBUS_TARGET register shall be as follow:

1. TCPC supplies vSafe5V over VBUS
2. TCPM writes to VBUS_NONDEFAULT_TARGET to set the target voltage level of COMMAND.SourceVbusNondefaultVoltage
3. TCPM issues COMMAND.SourceVbusNondefaultVoltage
4. TCPC starts the operation of transitioning to the target voltage level as given in VBUS_NONDEFAULT_TARGET. For TCPC that integrates the power converter, the TCPC shall control the voltage transitioning and meet the power supply requirements in the ***USB PD*** specification.

If the TCPM has a new target voltage level for COMMAND.SourceVbusNondefaultVoltage, it shall repeat Step2. The TCPM is not required to go back to vSafe5V and then to a different voltage. It may go directly to the new voltage by writing new values to VBUS_NONDEFAULT_TARGET and then issuing the COMMAND.SourceVbusNondefaultVoltage.
Figure 4-16 and Figure 4-17 indicate the flow for VBUS transition between vSafe5V and a nondefault voltage level.

Table 4-27. 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 and vSafe0V detection. This is an invalid COMMAND[2] if the TCPC has sourcing or sinking power over VBUS enabled.
		0011 0011b	**EnableVbusDetect**. Enable VBUS present and vSafe0V detection.
		0100 0100b	**DisableSinkVbus**. Disable sinking power over VBUS and discharge to vSave0V. This COMMAND does not disable POWER_STATUS.VBUSPresent detection. The TCPC shall clear FAULT_STATUS.InternalorExternalOCP and FAULT_STATUS.InternalorExternalOVP.
		0101 0101b	**SinkVbus**. Enable sinking power over VBUS and enable VBUS present detection. This is an invalid COMMAND[2] if the TCPC has sourcing power over Vbus enabled.
		0110 0110b	**DisableSourceVbus**. Disable sourcing power over VBUS and discharge to vSave0V. 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. The TCPC shall transition to vSafe5V if it is sourcing nondefault voltage. This is an invalid COMMAND[2] if the TCPC has sinking power over VBUS enabled.
		1000 1000b	**SourceVbusNondefaultVoltage**. Execute transitioning VBUS to a nondefault voltage level. This is an invalid COMMAND[2] if the TCPC is currently sinking voltage from VBUS, or the TCPC does not have the ability to source nondefault voltages (i.e. other than vSafe5V). This is also an invalid COMMAND[2] if the TCPC is not already sourcing power over Vbus. The target VBUS voltage to be sourced may be set in a vendor defined manner. The TCPM may need to write to vendor defined register 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. If ROLE_CONTROL.CC1/CC2 are not both 01b or 10b, or if POWER_CONTROL.AutoDischargeDisconnect is not 0b, then do not start toggling. 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 (as defined in ***USB Type-C***). This would result in ROLE_CONTROL staying the same.
		1010 1010b	**RxOneMore**. Configure the TCPC to automatically clear the RECEIVE_DETECT register after sending the next GoodCRC. This is used to disable message passing at a known point regardless of message separation or the depth of the RECEIVE_BUFFER in the TCPC.
		1100 1100b	**SendFRSwapSignal.** Source TCPC sends Fast Role Swap signal within tTCPCSendFRSwap after receiving this COMMAND if POWER_CONTROL.FastRoleSwapEnable = 1b. This is an invalid COMMAND[2] if POWER_CONTROL.FastRoleSwapEnable = 0b.

Bit(s)	Name	Register Setting	Description
		1101 1101b	**ResetTransmitBuffer.** The TCPC resets the pointer of the TRANSMIT_BUFFER register to offset 1 and the contents of TRANSMIT_BUFFER becomes invalid when this COMMAND is issued by the TCPM. This COMMAND shall be supported by TCPC compliant with USB Port Controller Specification Revision 2.0,
		1110 1110b	**ResetReceiveBuffer.** The TCPC resets the pointer of RECEIVE_BUFFER when this COMMAND is issued by the TCPM. If the pointer of RECEIVE_BUFFER.RX_BUF_BYTE_x is at 132 or less, writing this COMMAND would reset the pointer to 1. If the pointer of RECEIVE_BUFFER.RX_BUF_BYTE_x is at 133 or higher, writing this COMMAND would reset the pointer to 133. Refer to Sections 4.7.6 and 4.7.7. TCPC does not clear the content of the buffer upon receiving this COMMAND. The TCPM issues this COMMAND in order to re-read the RECEIVE_BUFFER.RX_BUF_BYTE_x. This COMMAND shall be supported by TCPC compliant with USB Port Controller Specification Revision 2.0,
		1111 1111b	**I2C Idle**

Note:
1. All other values are reserved; shall be ignored by the receiver
2. The TCPC shall ignore an invalid COMMAND and assert the I2CInterfaceError bit in the FAULT_STATUS register.

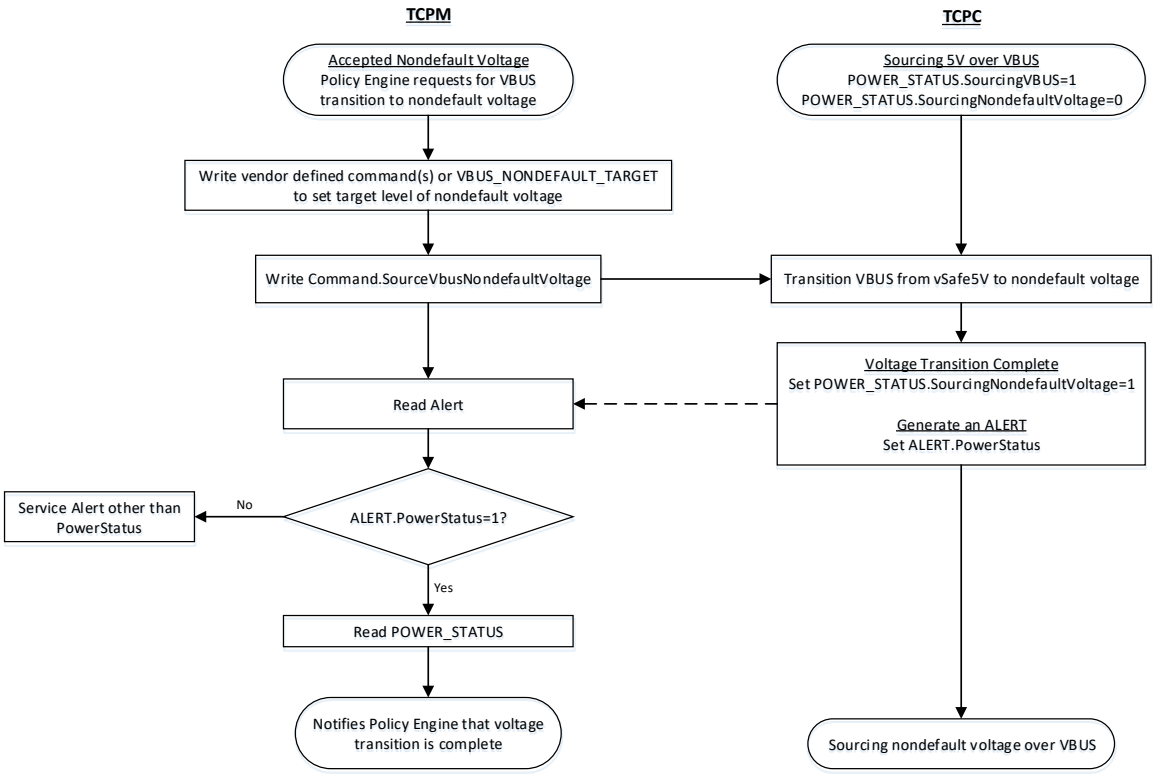


Figure 4-16. Transition from vSafe5V to Nondefault Voltage

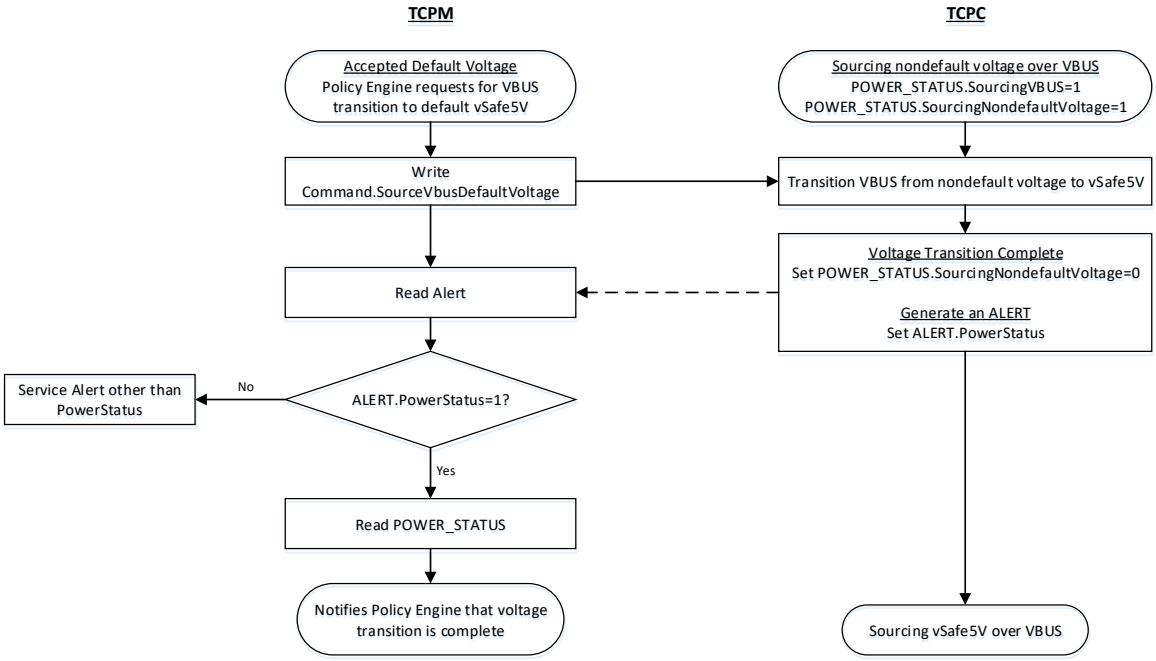


Figure 4-17. Transition from Nondefault Voltage to vSafe5V

4.4.9 Capability Registers

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

4.4.9.1 DEVICE_CAPABILITIES (Normative)

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

Table 4-28. DEVICE_CAPABILITIES_1 Register Definition

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

Bit(s)	Name	Description
B4	SOP'_DBG/SOP"_DBG Support	0b: All SOP* except SOP'_DBG/SOP"_DBG 1b: All SOP* messages are supported Configured in RECEIVE_DETECT and TRANSMIT
B3	Source VCONN	0b: TCPC is not capable of switching the VCONN Source 1b: TCPC is capable of switching the VCONN Source If this bit is set to 1b, POWER_CONTROL.EnableVCONN and POWER_STATUS.VCONNPresent shall be implemented.
B2	Sink VBUS	0b: TCPC is not capable controlling the sink path to the system load 1b: TCPC is capable of controlling the sink path to the system load If this bit is set to 1b, POWER_STATUS.SinkingVbus, COMMAND.SinkVbus, and COMMAND.DisableSinkVbus shall be implemented.
B1	Source Nondefault VBUS	0b: TCPC is not capable of sourcing nondefault voltages over VBUS 1b: TCPC is capable of sourcing nondefault voltages over VBUS If this bit is set to 1b, VBUS_VOLTAGE, POWER_STATUS.SourcingNondefaultVbus, and COMMAND.SourceVbusNondefaultVoltage shall be implemented. NOTE: DEVICE_CAPABILITIES_1.VbusMeasurementAlarmCapable shall be set to 1b if DEVICE_CAPABILITIES_1.SourceNondefaultVbus is set to 1b.
B0	Source VBUS	0b: TCPC is not capable of controlling the source path to VBUS 1b: TCPC is capable of controlling the source path to VBUS If this bit is set to 1b, POWER_STATUS.SourcingVbus, COMMAND.SourceVbusDefaultVoltage, COMMAND.DisableSourceVbus, COMMAND.EnableVbusDetect and COMMAND.DisableVbusDetect shall be implemented.

Table 4-29. DEVICE_CAPABILITIES_2 Register Definition

Bit(s)	Name	Description
B15	DEVICE_CAPABILITIES_3 Support	0b: TCPC does not support the DEVICE_CAPABILITIES_3 register. 1b: TCPC supports the DEVICE_CAPABILITIES_3 register.
B14	Message Disable Disconnect	0b: Sink TCPC disables PD message delivery when ALERT.VbusinkDisconnectDetected has been asserted 1b: Sink TCPC disables PD message delivery using the condition as defined in RECEIVE_DETECT.MessageDisableDisconnect If this bit is set to 1b, RECEIVE_DETECT.MessageDisableDisconnect shall be implemented.
B13	Generic Timer	0b: GENERIC_TIMER register is not supported 1b: GENERIC_TIMER register is supported
B12	Long Message	0b: TCPC only supports 30 bytes content of the SOP* message. The value in READABLE_BYTE_COUNT shall be less than or equal to 31. The value in I2C_WRITE_BYTE_COUNT shall be less than or equal to 30. 1b: TCPC is capable of supporting 264 bytes content of the SOP* message. The TRANSMIT_BUFFER holds up to 264 bytes content of the SOP* message. The TCPM can write up to 132 bytes to the TX_BUF_BYTE_x in one burst. The value supported in I2C_WRITE_BYTE_COUNT shall be up to 132. RECEIVE_BUFFER holds up to 264 bytes content SOP* message plus a 30 bytes content SOP* message. Refer to Sections 4.7.1, 4.7.2, 4.7.5, 4.7.6 and 4.7.7 for details on the message passing mechanisms.

Bit(s)	Name	Description
B11	SMBus PEC	0b: TCPC_CONTROL.EnableSMBusPEC not implemented 1b: TCPC_CONTROL.EnableSMBusPEC implemented
B10	Source FR Swap	0b: Not capable of sending Fast Role Swap signal as Source when receiving COMMAND.SendFRSwapSignal or receiving STANDARD INPUT Source Fast Role Swap low. 1b: Capable of sending Fast Role Swap signal as Source TCPC when receiving COMMAND.SendFRSwapSignal. If STANDARD_INPUT_CAPABITILIES.SourceFRSwap = 1b, capable of sending Fast Role Swap signal as Source when STANDARD INPUT Source Fast Role Swap is set low.
B9	Sink FR Swap	0b: POWER_CONTROL.FastRoleSwapEnable not supported as Sink 1b: POWER_CONTROL.FastRoleSwapEnable supported as Sink
B8	Watchdog Timer	0b: TCPC_CONTROL.EnableWatchdogTimer not implemented 1b: TCPC_CONTROL.EnableWatchdogTimer implemented
B7	Sink Disconnect Detection	0b: VBUS_SINK_DISCONNECT_THRESHOLD not implemented (default), use POWER_STATUS.VbusPresent=0b to indicate a Sink disconnect. 1b: VBUS_SINK_DISCONNECT_THRESHOLD implemented, VBUS falling below VBUS_SINK_DISCONNECT_THRESHOLD is used as indication of Sink disconnect.
B6	Stop Discharge Threshold	0b: VBUS_STOP_DISCHARGE_THRESHOLD not implemented (default) 1b: VBUS_STOP_DISCHARGE_THRESHOLD implemented
B5..4	VBUS Voltage Alarm LSB	The TCPC may ignore the number of Least Significant Bits (LSBs) of 4 Threshold and Alarm registers as denoted in this field as long as the TCPC provides the required accuracy for each threshold (see section 4.4.18). Whether or not the TCPC ignores these bits may depend on the instantaneous Voltage of VBUS. The TCPM can use this information to round up or down the affected threshold values as appropriate. The registers affected are: VBUS_SINK_DISCONNECT_THRESHOLD, VBUS_STOP_DISCHARGE_THRESHOLD, VBUS_VOLTAGE_ALARM_HI_CFG, and VBUS_VOLTAGE_ALARM_LO_CFG. 00: TCPC uses all allocated bits for its voltage and alarm thresholds for an effective threshold step of 25 mV 01: TCPC may ignore the least significant bit for an effective threshold step of 50 mV 10: TCPC may ignore the 2 least significant bits for an effective threshold step of 100 mV. 11: Reserved This field shall be ignored if VBUS_VOLTAGE_ALARM_LO_CFG, VBUS_VOLTAGE_ALARM_HI, VBUS_SINK_DISCONNECT_THRESHOLD, and VBUS_STOP_DISCHARGE_THRESHOLD are not supported

Bit(s)	Name	Description
B3..1	VCONN Power Supported	000b: 1.0W 001b: 1.5W 010b: 2.0W 011b: 3W 100b: 4W 101b: 5W 110b: 6W 111b: External
B0	VCONN Overcurrent Fault Capable	0b: TCPC is not capable of detecting a VCONN over-current fault 1b: TCPC is capable of detecting a VCONN over-current fault If this bit is set to 1b, FAULT_STATUS.VCONNOverCurrentFault and FAULT_CONTROL.VCONNOverCurrentFault shall be implemented.

4.4.9.2 DEVICE_CAPABILITIES_3 (Optional Normative)

This register is required if DEVICE_CAPABILITIES_2. DEVICE_CAPABILITIES_3 Supported is set to 1b as described in 4.4.9.1

The purpose of this register is to provide optional normative way to allow the TCPC to communicate additional capabilities beyond the DEVICE_CAPABILITIES_1 and DEVICE_CAPABILITIES_2 registers.

Table 4-30. DEVICE_CAPABILITIES_3 Register Definition

Bit(s)	Name	Description
B15..3	Reserved	Shall be set to zero by sender and ignored by receiver.
B2:0	VBUS Voltage Support	Describes the maximum nominal VBUS Voltage that can be supported by the TCPC. Support for the VBUS Voltage in this register means that the VBUS Voltage Measurement, VBUS Threshold and VBUS alarm registers are read by the TCPM as described by this specification and that the TCPC is designed to support VBUS Voltages up to this level. The TCPC shall be capable of supporting Voltages of at least the maximum nominal VBUS Support + 10% in the TCPC VBUS Threshold and Alarm registers: VBUS_SINK_DISCONNECT_THRESHOLD, VBUS_STOP_DISCHARGE_THRESHOLD, VBUS_VOLTAGE_ALARM_HI_CFG, and VBUS_VOLTAGE_ALARM_LO_CFG. 000b: 5 Volts* 001b: 9 Volts 010b: 15 Volts 011b: 20 Volts 100b: 28 Volts 101b: 36 Volts 110b: 48 Volts 111b: Reserved. *All voltages are Nominal

4.4.9.3 STANDARD_INPUT_CAPABILITIES (Normative)

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

Table 4-31. STANDARD_INPUT_CAPABILITIES Register Definition

Bit(s)	Name	Description
B7…5	Reserved	Shall be set to zero by sender and ignored by receiver
B4…3	Source Fast Role Swap	00b: Not present in TCPC 01b: Present in TCPC as an input only pin 10b: Present in TCPC as a bidirectional pin, sharing with the STANDARD OUTPUT SIGNAL VBUS Sink Disconnect Detect Indicator. The "VBUS Sink Disconnect Detect Indicator" bit in STANDARD_OUTPUT_CAPABILITIES register shall also be set to 1. 11b: Reserved
B2	VBUS External Over Voltage Fault	0b: Not present in TCPC 1b: Present in TCPC
B1	VBUS External Over Current Fault	0b: Not present in TCPC 1b: Present in TCPC
B0	Force Off VBUS (Source or Sink)	0b: Not present in TCPC 1b: Present in TCPC

4.4.9.4 STANDARD_OUTPUT_CAPABILITIES (Normative)

This register is in the nonvolatile memory of the TCPC. This register describes the optional normative 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.

Table 4-32. STANDARD_OUTPUT_CAPABILITIES Register Definition

Bit(s)	Name	Description
B7	VBUS Sink Disconnect Detect Indicator	0b: Not present in TCPC 1b: Present in TCPC Shall present in TCPC if "Source Fast Role Swap" in STANDARD_INPUT_CAPABILITIES is set to 10b (present as a bidirectional pin).
B6	Debug Accessory Indicator	0b: Not present in TCPC 1b: Present in TCPC If this bit is not set, the TCPC does not support Debug Accessory State-Machine as shown in Figure 4-21, Figure 4-22 and Figure 4-23.
B5	VBUS Present Monitor	0b: Not present in TCPC 1b: Present in TCPC
B4	Audio Adapter Accessory Indicator	0b: Not present in TCPC 1b: Present in TCPC
B3	Active Cable Indicator	0b: Not present in TCPC 1b: Present in TCPC

Bit(s)	Name	Description
B2	MUX Configuration Control	0b: Not present in TCPC 1b: Present in TCPC
B1	Connection Present	0b: No Connection 1b: Connection Controlled by the TCPM.
B0	Connector Orientation	0b: Not present in TCPC 1b: Present in TCPC

4.4.10 CONFIGURE EXTENDED1 (Optional Normative)

The TCPM writes to this register to configure the extended functions.

Table 4-33. CONFIG_EXTENDED1 Register Definition

Bit(s)	Name	Description
B7..2	Reserved	Shall be set to zero by sender and ignored by receiver
B1	FR Swap Bidirectional Pin	0b: The bidirectional pin is configured as STANDARD INPUT SIGNAL Source Fast Role Swap (default) 1b: The bidirectional pin is configured as STANDARD OUTPUT SIGNAL VBUS Sink Disconnect Detect Indicator The TCPC shall ignore a write to this bit and assert FAULT_STATUS.I2CInterfaceError if STANDARD_INPUT_CAPABILITIES.SourceFastRoleSwap is not set to 10b (present as a bidirectional pin), and STANDARD_OUTPUT_CAPABILITIES.VbusSinkDisconnectDetectIndicator is not set to 1.
B0	Standard Input Source FR Swap	0b: Allow STANDARD INPUT SIGNAL Source Fast Role Swap to trigger sending Fast Role Swap signal (default) 1b: Block STANDARD INPUT SIGNAL Source Fast Role Swap to trigger sending Fast Role Swap signal This bit enables or disables STANDARD INPUT SIGNAL Source Fast Role Swap functionality. Required if STANDARD_INPUT_CAPABILITIES.SourceFastRoleSwap is set to either 01b or 10b (present).

4.4.11 GENERIC_TIMER (Optional Normative)

The TCPM writes a non-zero value to this register to start the general purpose timer. If the TCPM writes a non-zero value to this register before the timer has expired, the timer is restarted with the last written non-zero value. After the timer has expired, the timer does not restart until TCPM writes a non-zero value to this register. The timer shall stop when a zero value is written to this register.
ALERT_EXTENDED.TimerExpired is asserted when the last written non-zero timer value has expired. Clearing the ALERT_EXTENDED.TimerExpired has no effect to this register. To avoid a race condition, the TCPM may write a zero value to this register before writing a non-zero value to start the timer.

Table 4-34. GENERIC_TIMER Register Definition

Bit(s)	Name	Description
B15...0	GENERIC_TIMER	16-bit timer value with 0.1ms LSB. A non-zero value starts the timer. A value of zero stops the timer. The timer does not restart after it has expired. Required if DEVICE_CAPABILITIES_2.GenericTimer = 1b.

4.4.12 MESSAGE_HEADER_INFO (Normative)

The TCPC shall set this register at power on per Table 4-17. 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 4-35. 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 1b: Message originated from a Cable Plug
B3	Data Role	0b: UFP 1b: DFP
B2..1	***USB PD*** Specification Revision	00b: Revision 1.0 01b: Revision 2.0 10b: Revision 3.0 11b: Reserved
B0	Power Role	0b: Sink 1b: Source

4.4.13 RECEIVE_DETECT (Normative)

Set by TCPM, cleared by TCPM, or the 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. The bits must be written to be enabled. When the TCPM sets all bits in RECEIVE_DETECT to zero, the TCPC shall disable automatic transmission of GoodCRC message and discard COMMAND.RxOneMore if it has not been acted upon. The TCPM may power down PD message delivery per Section 4.8.2.
The TCPC shall be capable of receiving a CableReset if DEVICE_CAPABILITIES_1.PowerRolesSupported =101b, the TCPC may be capable of receiving a CableReset if DEVICE_CAPABILITIES_1.PowerRolesSupported != 101b.
The TCPC shall disable PD message delivery under the following conditions:

- When RECEIVE_DETECT.HardReset is set and a Hard Reset is received
- When the TCPM writes to TRANSMIT to request a Hard Reset transmission
- When RECEIVE_DETECT.CableReset is set and a Cable Reset is received
- On detection of a disconnect (Sink TCPC shall use condition as defined in RECEIVE_DETECT.MessageDisableDisconnect)

The following happens when the TCPC disables PD message delivery:

- The TCPC disables automatic transmission of GoodCRC message.
- The TCPC shall discard COMMAND.RxOneMore if it has not been acted upon.
- The TCPC sets all bits in RECEIVE_DETECT to zero
- The TCPC sets all bits in READABLE_BYTE_COUNT to zero

Transmitting a Cable Reset by the TCPC has no effect on RECEIVE_DETECT register.

Table 4-36. RECEIVE_DETECT Register Definition

Bit(s)	Name	Description
B7	Message Disable Disconnect	0b: Sink TCPC shall disable PD message delivery when ALERT.VbusSinkDisconnectDetected has been asserted i.e. VBUS Sink disconnect is detected (default) 1b: Sink TCPC shall disable PD message delivery when SNK.Open (Sink CC pin is below maximum vRa) is detected for at least tPDDebounce min (10ms). Regardless of the setting of this bit, Source TCPC always uses SRC.Open as an indication of disconnect to disable PD message delivery. The TCPC clears the RECEIVE_DETECT and READABLE_BYTE_COUNT registers to disable the PD message delivery. Required if DEVICE_CAPABILITIES_2.MessageDisableDisconnect = 1b
B6	Enable Cable Reset	0b: TCPC does not detect Cable Reset signaling (default) 1b: TCPC detects Cable Reset signaling
B5	Enable Hard Reset	0b: TCPC does not detect Hard Reset signaling (default) 1b: TCPC detects Hard Reset signaling
B4	Enable SOP_DBG'' message	0b: TCPC does not detect SOP_DBG'' message (default) 1b: TCPC detects SOP_DBG'' message
B3	Enable SOP_DBG' message	0b: TCPC does not detect SOP_DBG' message (default) 1b: TCPC detects SOP_DBG' message
B2	Enable SOP'' message	0b: TCPC does not detect SOP'' message (default) 1b: TCPC detects SOP'' message
B1	Enable SOP' message	0b: TCPC does not detect SOP' message (default) 1b: TCPC detects SOP' message
B0	Enable SOP message	0b: TCPC does not detect SOP message (default) 1b: TCPC detects SOP message

4.4.14 RECEIVE_BUFFER (Normative)

The RECEIVE_BUFFER comprises of three sets of registers: READABLE_BYTE_COUNT, RX_BUF_FRAME_TYPE and RX_BUF_BYTE_x. These registers can only be accessed by reading at a common register address 30h (refer to Table 4-1 and Figure 4-6). These registers indicate the status of the received SOP* message buffer. These registers shall be read by the TCPM when the TCPC indicates a SOP* message was received in the Alert Status registers. The TCPM reads the READABLE_BYTE_COUNT to determine the number of bytes in the RX_BUF_BYTE_x. The TCPM reads the RX_BUF_FRAME_TYPE to determine the type of message. The TCPM then reads the content of the ***USB PD*** message in RX_BUF_BYTE_x.
The TCPC shall set the READABLE_BYTE_COUNT to 0 after the interrupt has been cleared.
See Sections 4.7.5, 4.7.6, 4.7.7, 4.7.8, and 4.7.9 for information on receiving SOP* messages, Hard Reset, and Cable Reset messages respectively.
The TCPC shall disable PD message delivery and set the READABLE_BYTE_COUNT to zero upon detection of a disconnect (Sink TCPC shall use the disconnect condition as defined in RECEIVE_DETECT.MessageDisableDisconnect). The TCPM power down PD message delivery per Section 4.8.2.
RECEIVE_BUFFER is read only.
If DEVICE_CAPABILITIES_2.LongMessage is set to zero, the RECEIVE_BUFFER is sized to hold two 30 bytes SOP* messages.
If DEVICE_CAPABILITIES_2.LongMessage is set to one, the RECEIVE_BUFFER is sized to hold a 264 bytes SOP* message plus a 30 bytes SOP* message.

Regardless of the DEVICE_CAPABILITIES_2.LongMessage setting, TCPC shall automatically increment the pointer of RX_BUF_BYTE_x as TCPM reads RX_BUF_BYTE_x. TCPM can re-read RX_BUF_BYTE_x at a zero offset by writing to COMMAND.ResetReceiveBuffer (0xEE). However, the pointer of RX_BUF_BYTE_x would not increment if TCPM reads READABLE_BYTE_COUNT or RX_BUF_FRAME_TYPE.

Table 4-37. READABLE_BYTE_COUNT Definition

Bit(s)	Name	Description
B7…0	READABLE_BYTE_COUNT	Indicates the number of bytes in the RX_BUF_BYTE_x registers plus one (for the RX_BUF_FRAME_TYPE). The content of this register is undefined when the RECEIVE_BUFFER is cleared. If DEVICE_CAPABILITIES_2.LongMessage = 0, the value in this register shall be less than or equal to 31. If DEVICE_CAPABILITIES_2.LongMessage = 1, the value supported in this register shall be up to 133.

Table 4-38. RX_BUF_FRAME_TYPE Definition

Bit(s)	Name	Description
B7…3	Reserved	Shall be set to zero by sender and ignored by receiver
B2…0	Received SOP* Message	000b: Received SOP 001b: Received SOP' 010b: Received SOP'' 011b: Received SOP_DBG' 100b: Received SOP_DBG'' 110b: Received Cable Reset All others are reserved.

4.4.15 TRANSMIT (Normative)

The TCPM writes to this register to transmit a SOP* message where the SOP* message payload (i.e. the header bytes and the data bytes) was written into the TCPC's internal transmit buffer using the TRANSMIT_BUFFER register. The TCPC transmits the aggregate of data written to the TRANSMIT_BUFFER since the pointer was last reset either due to the TCPM writing to the TRANSMIT register or the TCPM writing to COMMAND.ResetTransmitBuffer (0xDD). The entire register shall be written at once and then sent. The TCPC shall clear the TRANSMIT register I2C_WRITE_BYTE_COUNT and its internal transmit buffer after executing the transmission regardless of the outcome (either successful, failed or discarded).
If the TCPM writes to TRANSMIT requesting a transmission that is not Hard Reset, Cable Reset or BIST Carrier Mode 2 (i.e. TRANSMIT.SOP*Message > 100b) and there are less than 2 bytes in the TX_BUF_BYTE_x register (i.e. the transmit buffer pointer is less than offset 3), the TCPC shall generate a FAULT_STATUS.I2CInterfaceError.
The TCPM shall require no message retry when transmitting a Hard Reset, a Cable Reset, or a BIST Carrier Mode 2 signaling. The TCPC shall ignore the Retry Counter bits (B5…4) when transmitting a Hard Reset, a Cable Reset, or a 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 ***USB 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*MessageStatus = 1 the TCPC shall discard the transmit request and assert ALERT.TransmitSOP*MessageDiscarded.
The TCPC shall clear the RECEIVE_DETECT and the READABLE_BYTE_COUNT register to disable the ***USB PD*** message passing when the TCPM writes the TRANSMIT register requesting a Hard Reset.
The TCPC shall assert one and only one of ALERT.TransmitSuccessful, TransmitFailed, or TransmitDiscarded following a write of the TRANSMIT register except when a Hard Reset or a Cable Reset is transmitted. The TCPC shall assert both ALERT.TransmitSuccessful and ALERT.TransmitFailed after it completes the sending of a Hard Reset or a Cable Reset.
After the TCPC has transmitted the BIST Carrier Mode signaling and exited BIST Carrier Mode, the TCPC shall generate either ALERT.TransmitSuccessful (if successfully sent) or ALERT.TransmitDiscarded (if not successfully sent due to an incoming received message).

Table 4-39. 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 01b: Automatically retry message transmission once 10b: Automatically retry message transmission twice 11b: Automatically retry message transmission three times
B3	Reserved	Shall be set to zero by sender, shall be ignored by receiver
B2..0	Transmit SOP* message	000b: Transmit SOP 001b: Transmit SOP' 010b: Transmit SOP'' 011b: Transmit SOP_DBG' 100b: Transmit SOP_DBG'' 101b: Transmit Hard Reset 110b: Transmit Cable Reset 111b: Transmit BIST Carrier Mode 2 (TCPC shall exit the BIST mode no later than tBISTContMode max)

4.4.16 TRANSMIT_BUFFER (Normative)

The TRANSMIT_BUFFER holds the I2C_WRITE_BYTE_COUNT and the portion of the SOP* ***USB PD*** message payload (including the header and/or the data bytes) most recently written by the TCPM in TX_BUF_BYTE_x. TX_BUF_BYTE_x is "hidden" and can only be accessed by writing to register address 51h (refer to Table 4-1 and Figure 4-5).
If DEVICE_CAPABILITIES_2.LongMessage is set to zero, the TRANSMIT_BUFFER is capable of holding 30 byte SOP* message. The TCPM can write up to 30 bytes to the TX_BUF_BYTE_x in one burst.

If DEVICE_CAPABILITIES_2.LongMessage is set to one, TRANSMIT _BUFFER is capable of holding 264 byte SOP* message. The TCPM can write up to 132 bytes to the TX_BUF_BYTE_x in one burst.
Regardless of the DEVICE_CAPABILITIES_2.LongMessage setting, the TCPC automatically increments the TX_BUF_BYTE_x offset as TCPM writes to TX_BUF_BYTE_x. The TCPM can re-write to TX_BUF_BYTE_x beginning at offset 1 by writing to COMMAND.ResetTransmitBuffer.
The TCPM shall write as many bytes in the buffer as defined in the I2C_WRITE_BYTE_COUNT in one I2C write transaction. If the I2C_WRITE_BYTE_COUNT is different than the number of bytes written in the buffer, the TCPC shall assert FAULT_STATUS.I2CInterfaceError and ignore the write (i.e. no change in the TX_BUF_BYTE_x content and the offset).

Table 4-40. I2C_WRITE_BYTE_COUNT Definition

Bit(s)	Name	Description
B7..0	I2C_WRITE_BYTE_COUNT	The number of bytes the TCPM intends to write to the TX_BUF_BYTE_x in the given I2C/SMBus transaction. The TCPM shall write as many bytes in the buffer as defined in this register in one I2C write transaction. If DEVICE_CAPABILITIES_2.LongMessage = 0, the TCPC shall ignore the I2C transaction if I2C_WRITE_BYTE_COUNT is more than 30. If DEVICE_CAPABILITIES_2.LongMessage = 1, the TCPC shall ignore the I2C transaction if I2C_WRITE_BYTE_COUNT is more than 132.

4.4.17 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-C*** Connector. The TCPC shall maintain synchronization between the upper and lower 8 bits of the register.
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.
This register is required if the TCPC is capable of sourcing or sinking nondefault VBUS voltage (i.e. other than vSafe5V). This register is required if it is reported as supported in one of the DEVICE_CAPABILITES registers, Section 4.4.9.1.

Table 4-41. 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. 01: VBUS measurement divided by 2 10: VBUS measurement divided by 4 11: reserved
B9…0	VBUS voltage measurement	10-bit measurement of (VBUS / Scale Factor) TCPM multiplies this value by the scale factor to obtain the voltage measurement. All voltages shall meet +/-2% absolute value or +/-50mV, whichever is greater. The LSB is 25mV.

4.4.18 Voltage Thresholds

These registers control the VBUS voltage thresholds. The registers are set by the TCPM and read by the TCPC. Each threshold has a minimum accuracy requirement that the TCPC shall support. The TCPC may ignore the low-order bits of each field as declared by DEVICE_CAPABILITIES_2.VBUSVoltageAlarmLSB only when the TCPC can do so while providing the minimum accuracy of the corresponding threshold.

4.4.18.1 VBUS _SINK_DISCONNECT_THRESHOLD

This register defines an edge-triggered threshold. This register is required by TCPCs which act as a Sink and are capable of sinking power through nondefault VBUS voltages (i.e. voltages other than vSafe5V). Implementation of this register shall be declared in DEVICE_CAPABILITIES_2.SinkDisconnectDetection.
This register is optional normative for TCPCs acting as Source only. This register has no action for a Source.
The TCPM writes to this register to set the threshold at which a Sink will start the Automatic Sink Discharge (Section 4.4.5.4.2) if it is in the Attached_Snk state as shown in Figure 4-19 and Figure 4-20.

Table 4-42. VBUS_SINK_DISCONNECT_THRESHOLD Register Description

Bit(s)	Name	Description
B15..12	Reserved	Shall be set to 0
B11..0	Voltage trip point	12-bit for voltage threshold with 25mV LSB. (Default 3.5V) +/- 5% accuracy. A value of zero disables this threshold The TCPC may ignore B0 or B1&B0 depending upon DEVICE_CAPABILITIES_2.VbusVoltageAlarmLsb.*

*TCPM FW DEVELOPER NOTE: *Number of bits and the resolution allow to have a voltage range much bigger than the voltage range in the **USB PD** specification. However, the alarm registers shall be programmed to not exceed the OVP limits specified in the **USB PD** specification that the TCPC is trying to comply with.*

4.4.18.2 VBUS_STOP_DISCHARGE_THRESHOLD

This register defines an edge-triggered threshold for the Force Discharge circuit. This register is optional normative. This register is required by TCPCs which act as a Source and are capable of sourcing nondefault VBUS voltages (i.e. voltages other than vSafe5V).
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 circuit when POWER_CONTROL.ForceDischarge = 1b. The TCPC shall not re-apply the Force Discharge circuit if VBUS rises above VBUS_STOP_DISCHARGE_THRESHOLD. A Source TCPC which does not support this register shall stop the VBUS discharge at to vSafe0V (max).
A TCPC acting as a Source shall always discharge to vSafe0V upon a disconnect, Hard Reset, or Power Role Swap.
This register is optional normative for TCPCs which act as a Sink. This register has no action for a Sink. On a disconnect, a Sink TCPC discharges VBUS to vSafe0V as described in Section 4.4.5.4.2. Implementation of this register shall be declared in DEVICE_CAPABILITIES_2.StopDischargeThreshold.

Table 4-43. VBUS_STOP_DISCHARGE_THRESHOLD Register Description

Bit(s)	Name	Description
B15…12	Reserved	Shall be set to 0
B11…0	Voltage trip point	12-bit for voltage threshold with 25mV LSB. (Default 0.8V) +/- 5% accuracy. The TCPC may ignore B0 or B1&B0 depending upon DEVICE_CAPABILITIES_2.VbusVoltageAlarmLsb.*

*TCPM FW DEVELOPER NOTE: *Number of bits and the resolution allow to have a voltage range much bigger than the voltage range in the **USB PD** specification. However, the alarm registers shall be programmed to not exceed the OVP limits specified in the **USB PD** specification that the TCPC is trying to comply with*.

4.4.18.3 Voltage Alarms (Optional Normative)

These registers define the level triggered alarm thresholds. These registers are required if DEVICE_CAPABILITIES_1.VBUSMeasurementAlarmCapable is set to 1b as described in Section 4.4.9.1.
Voltage alarms are required by TCPCs that handles nondefault voltages (i.e. voltages other than vSafe5V).
The TCPM can write to POWER_CONTROL.DisableVoltageAlarms = 1b to disable the voltage alarms. The TCPM writes to VBUS_VOLTAGE_ALARM_HI_CFG to set the high voltage alarm level. The TCPC sets ALERT.VBUSVoltageAlarmHi to 1 when VBUS exceeds the high voltage alarm level. The TCPC shall re-assert ALERT.VBUSVoltageAlarmHi if the high voltage condition on VBUS prevails after the TCPM has cleared ALERT.VBUSVoltageAlarmHi unless the TCPM disables the voltage alarms by setting POWER_CONTROL.DisableVoltageAlarms to 1b.

Table 4-44. VBUS_VOLTAGE_ALARM_HI_CFG Register Description

Bit(s)	Name	Description
B15…12	Reserved	Shall be set to 0
B11…0	Voltage trip point	12-bit for voltage threshold with 25mV LSB. +/- 5% accuracy. The TCPC may ignore B0 or B1&B0 depending upon DEVICE_CAPABILITIES_2.VBUSVoltageAlarmLsb.*

*TCPM FW DEVELOPER NOTE: *Number of bits and the resolution allow to have a voltage range much bigger than the voltage range in the **USB PD** specification. However,* the alarm registers shall be programmed to not exceed the OVP limits specified in the ***USB PD** specification that the TCPC is trying to comply with.*

The TCPM writes to VBUS_VOLTAGE_ALARM_LO_CFG to set the low voltage alarm level. The TCPC sets ALERT.VBUSVoltageAlarmLo to 1 when VBUS drops below the low voltage alarm level. The TCPC shall re-assert ALERT.VBUSVoltageAlarmLo if the low voltage condition on VBUS prevails after TCPM has cleared ALERT.VBUSVoltageAlarmLo unless the TCPM disables the voltage alarms by setting POWER_CONTROL.DisableVoltageAlarms to 1b.

Table 4-45. VBUS_VOLTAGE_ALARM_LO_CFG Register Description

Bit(s)	Name	Description
B15…12	Reserved	Shall be set to 0
B11…0	Voltage trip point	12-bit for voltage threshold with 25mV LSB. +/- 5% accuracy. The TCPC may ignore B0 or B1&B0 depending upon DEVICE_CAPABILITIES_2.VbusVoltageAlarmLsb.*

*TCPM FW DEVELOPER NOTE: *Number of bits and the resolution allow to have a voltage range much bigger than the voltage range in the **USB PD** specification. However,* the alarm *registers shall be programmed to not exceed the OVP limits specified in the **USB PD** specification that the TCPC is trying to comply with*.

4.4.19 VBUS_NONDEFAULT_TARGET (Optional Normative)

This register is required if DEVICE_CAPABILITIES_1.VBUSNondefaultTarget is set to 1b as described in Section 4.4.9.1.
The purpose of this register is to provide an optional normative way to set the target voltage for sourcing nondefault voltage over VBUS. The TCPM may write to this register to set the target voltage level of COMMAND.SourceVBUSNondefaultVoltage. For TCPC that integrates the power converter, the TCPC shall control the voltage transitioning and meet the power supply requirements in the **USB PD** specification.

Table 4-46. VBUS_NONDEFAULT_TARGET Register Description

Bit(s)	Name	Description
B15…0	VBUS nondefault voltage target	16-bit for the VBUS voltage target with 20mV LSB. +/- 5% accuracy. The TCPC shall ignore TCPM writing to this register if the value is less than 3.3V (A5h) or DEVICE_CAPABILITIES_1.VBUSNondefaultTarget is set to 0b.*

*TCPM FW DEVELOPER NOTE: *Number of bits and the resolution allow to have a voltage range much bigger than the voltage range in the **USB PD** specification. However, this register shall be programed to not exceed maximum nominal voltage defined in the DEVICE_CAPABILITIES_3.VBUS Voltage Support field if DEVICE_CAPABILITIES_2. DEVICE_CAPABILITIES_3 Supported is set to 1b. Otherwise, this register shall be programmed to not exceed allowable maximum nominal voltage specified in the **USB PD** specification and the range that the TCPC is trying to comply with. Example: Per PD3.1 maximum nominal voltage for SPR is 21V, and EPR is 48V.*

4.4.20 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.

4.5 STANDARD IO SIGNALS

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

4.5.1 STANDARD INPUT SIGNALS (Optional Normative)

Support for any of these signals shall be declared in the STANDARD_INPUT_CAPABILITIES register (Section 4.4.9.3). This section defines the STANDARD INPUT SIGNALs to the TCPC. Some of the input signals provide updates to the FAULT_STATUS register (Section 4.4.6.3). The TCPC shall set the FAULT_STATUS or ALERT_EXTENDED register based on the input level at the pin.

Table 4-47. STANDARD INPUT SIGNALs

Inputs	Type
VBUS External Over Current Fault	Low: Set STANDARD_INPUT Register to 0. No Over Current Fault High: Set STANDARD_INPUT Register to 1. Over Current Fault present Reported in FAULT_STATUS.InternalExternalOCP
VBUS External Over Voltage Fault	Low: Set STANDARD_INPUT Register to 0. No Over Voltage Fault. High: Set STANDARD_INPUT Register to 1. Over Voltage Fault present. Reported in FAULT_STATUS.InternalExternalOVP
Force Off VBUS (Source or Sink)	Low: Set STANDARD_INPUT Register to 0. Do not force VBUS off. High: Set STANDARD_INPUT Register to 1. Force VBUS Off. Reported in FAULT_STATUS.ForceOffVBUS
Source Fast Role Swap	High->Low: Send Fast Role Swap signal within tTCPCSendFRSwap. TCPC shall generate a bit-level NAK to the I2C write and then assert FAULT_STATUS.I2CInterfaceError when TCPC is transmitting the Fast Role Swap signal (as triggered by the STANDARD INPUT SIGNAL). Low->High: Reset Reported in ALERT_EXTENDED.SourceFRSwap

4.5.2 STANDARD OUTPUT SIGNALS (Optional Normative except Alert#)

Support for any of these signals shall be declared in the STANDARD_OUTPUT_CAPABILITIES register (Section 4.4.9.4). 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 4-48.
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 4-48. 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. High: No Alert Open Drain

Output	Type
Debug Accessory Connected#	High: No Debug Accessory connected Low: Debug Accessory connected Push/Pull or Open Drain
VBUS Present#	Low: VBUS is present High: VBUS is not present Push/Pull or Open Drain
Audio Accessory Connected#	High: No Audio Accessory connected Low: Audio Accessory connected Push/Pull or Open Drain
Active Cable Connected	High: Active Cable connected Low: No Active Cable connected Push/Pull or Open Drain
MUX Control 1	Low: No DP Alternate Mode High: DP Alternate Mode Push/Pull or Open Drain
MUX Control 0	Low: No connection or No USB Connection High: USB Connection Push/Pull or Open Drain
Connection Present	Low: No Connection High: Connection Push/Pull or Open Drain
Connector Orientation	Low: Normal (default) High: Flipped Push/Pull or Open Drain
VBUS Sink Disconnect Detected	Low: The TCPC indicates VBUS Sink disconnect has been detected and ALERT.VbusSinkDisconnectDetected has been asserted. This is either when POWER_STATUS.VbusPresent transitions from 1b to 0b (if DEVICE_CAPABILITIES_2.SinkDisconnectDetection=0b) or the TCPC detects VBUS falling below VBUS_SINK_DISCONNECT_THRESHOLD (if DEVICE_CAPABILITIES_2.SinkDisconnectDetection=1b). High: No VBUS Sink disconnect has been detected or the TCPC has cleared the status of this signal by writing a logical 1 to ALERT.VbusSinkDisconnectDetected Push/Pull or Open Drain

4.6 Type-C Port Controller Connection State Diagrams and Flows

Refer to Section 3.5 for the structure of the state diagram figures.
This section defines the behavior of a TCPC when using the DRP functionality.
This section provides two reference connection state machines for a DRP TCPC, namely the Main State-Machine (as shown in Figure 4-19 and Figure 4-20) and the Debug Accessory State-Machine (as shown in Figure 4-21, Figure 4-22 and Figure 4-23).
For a DRP TCPC that supports DebugAccessoryIndicator, both Main State-Machine and Debug Accessory State-Machine operate simultaneous.
As shown in Figure 4-18, after the TCPC has completed the power on, the Main State-Machine enters Maintain_State and the Debug Accessory State-Machine enters Start_Debug_Accesory state if DebugAccessoryIndicator is supported.
If the DRP TCPC supports DebugAccessoryIndicator, TCPM shall set TCPC_CONTROL.DebugAccessoryControl to 1 to prevent the TCPC sending COMMAND.Look4Connection to itself when the Debug Accessory State-Machine enters Start_Debug_Accessory.
When the Main State-Machine has entered Connected_Invalid state, the TCPC shall assert FAULT_STATUS.I2CInterfaceError and take the following recommended actions:

- The TCPC may disable sourcing power over V_{BUS}
- The TCPC may drive CC1 and CC2 pins as per ROLE_CONTROL register

Figure 4-24, Figure 4-25 and Figure 4-26 show flowcharts of a DRP TCPC transitioning between connection and disconnection.

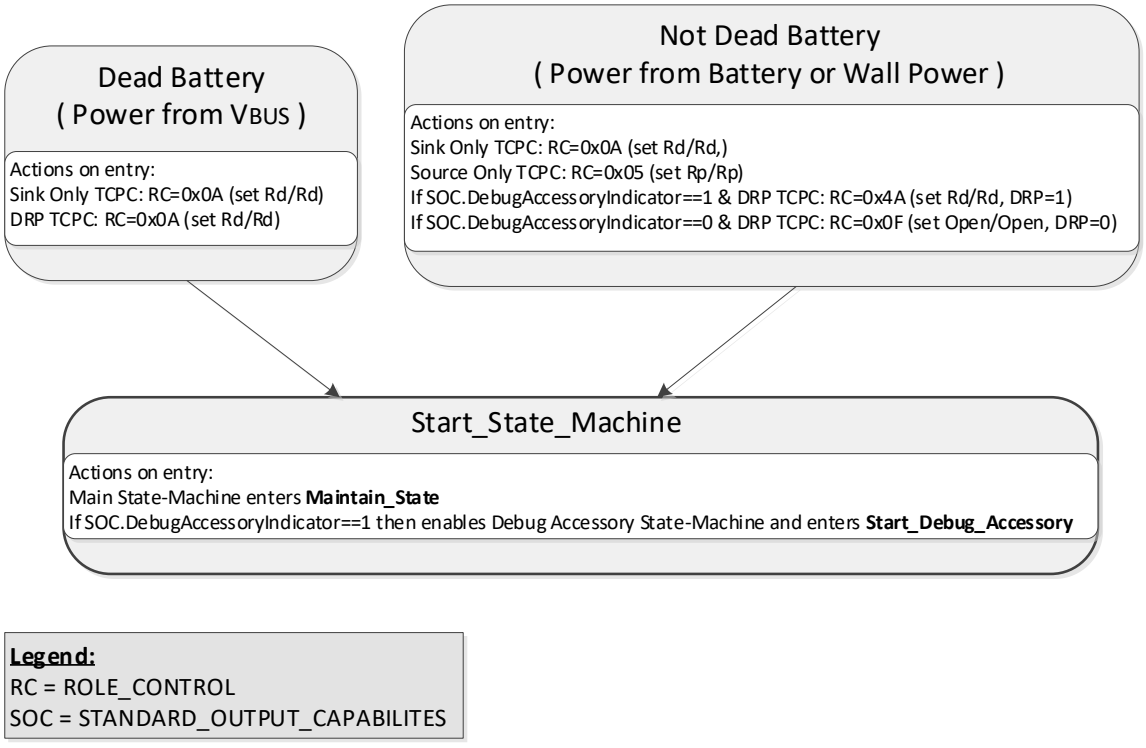


Figure 4-18. TCPC Power-On State Diagram

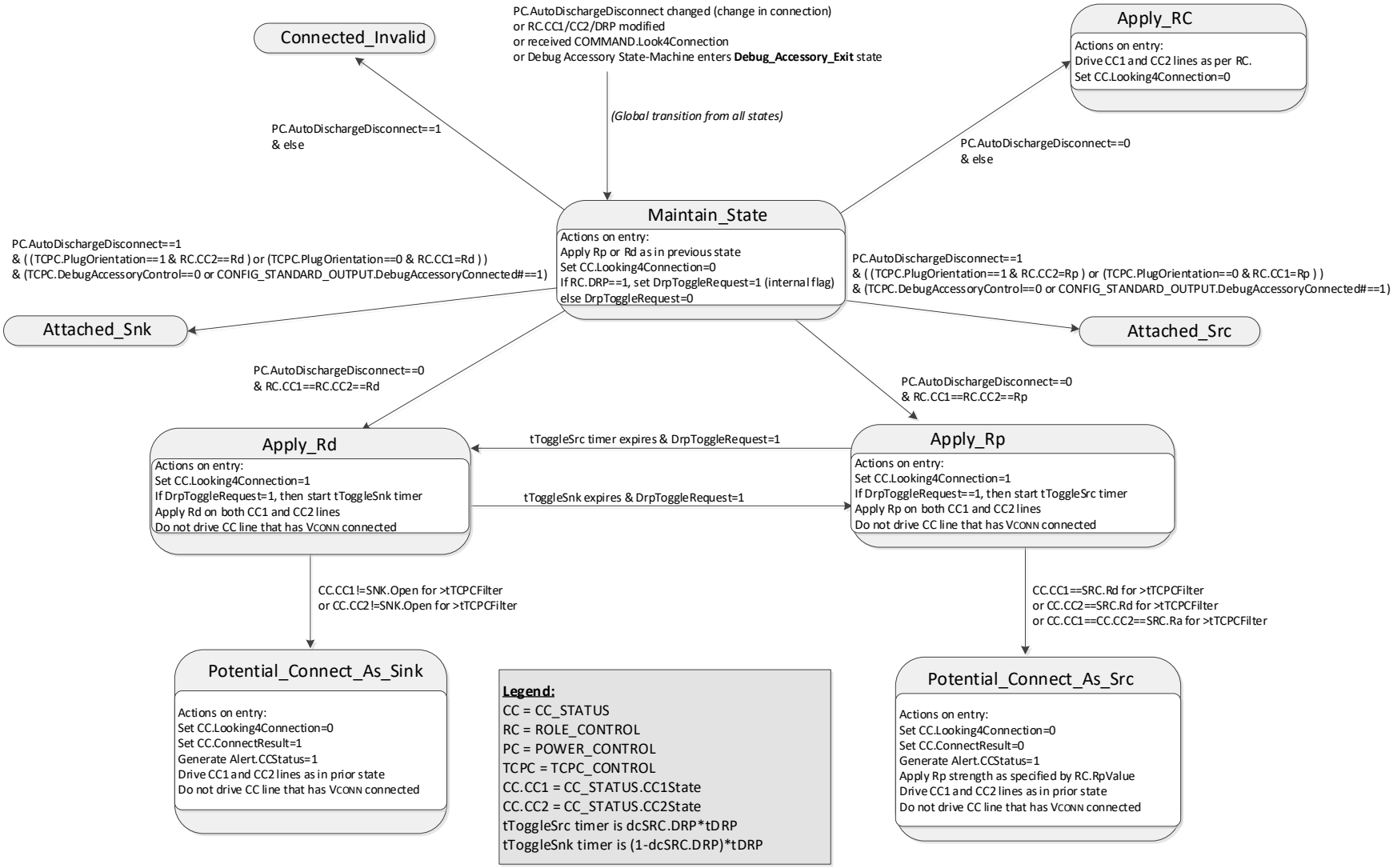


Figure 4-19. TCPC Main State-Machine: Before a Connection

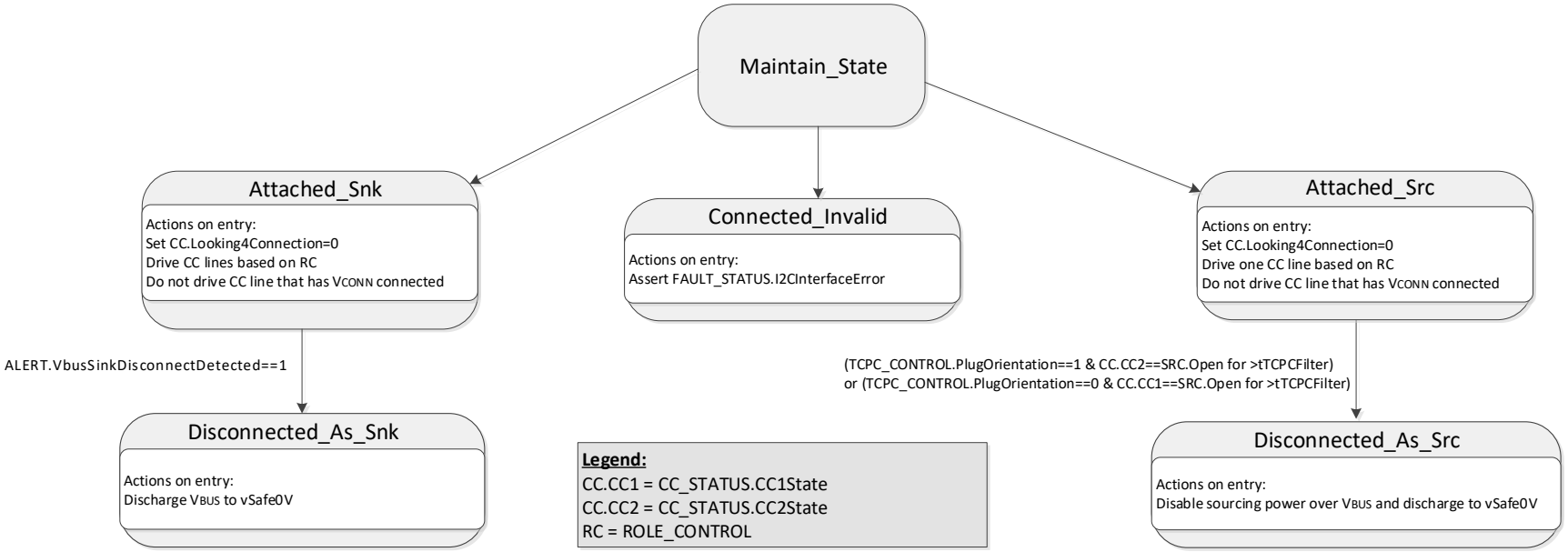


Figure 4-20. TCPC Main State-Machine: After a Connection

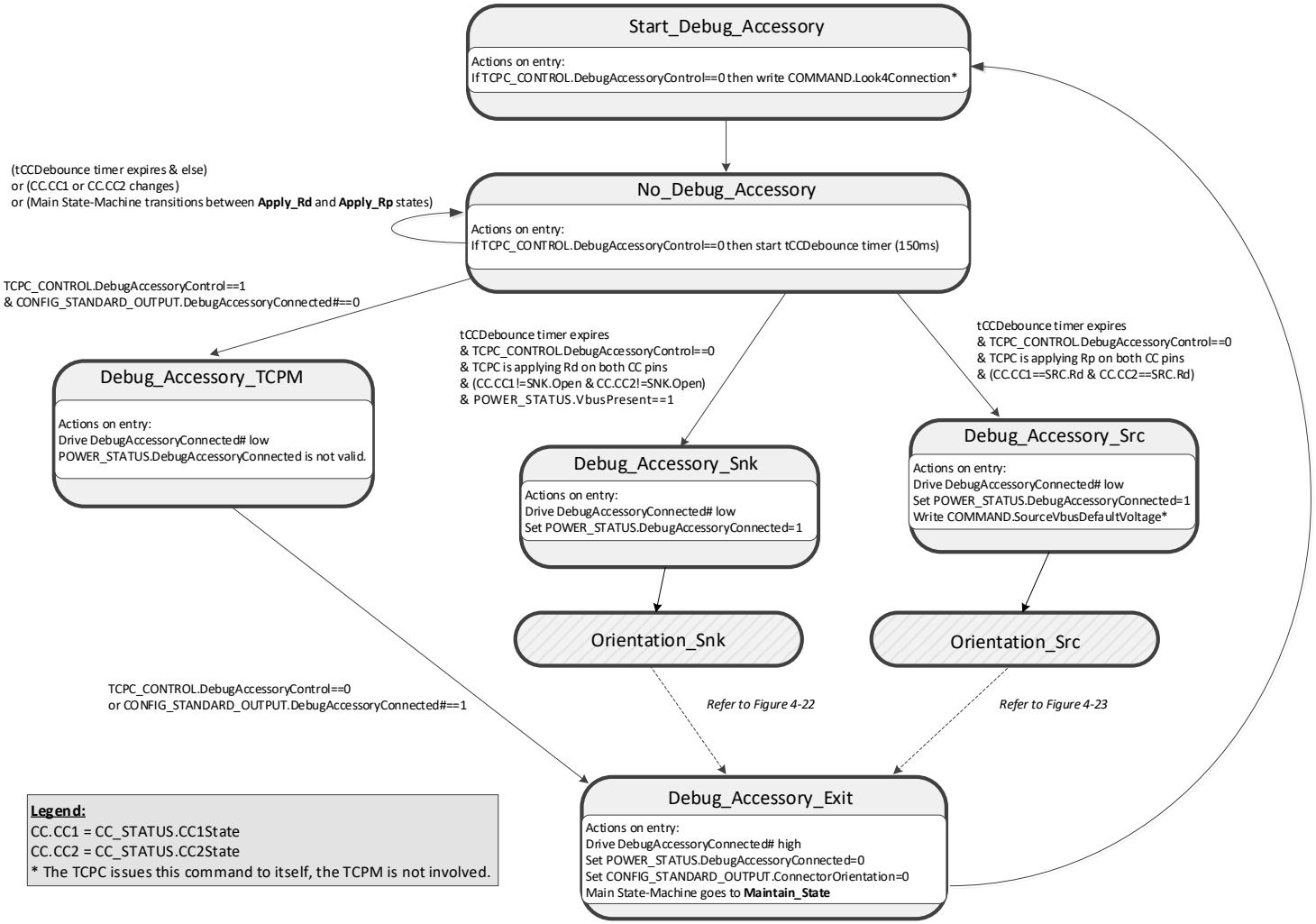


Figure 4-21. TCPC Debug Accessory State-Machine

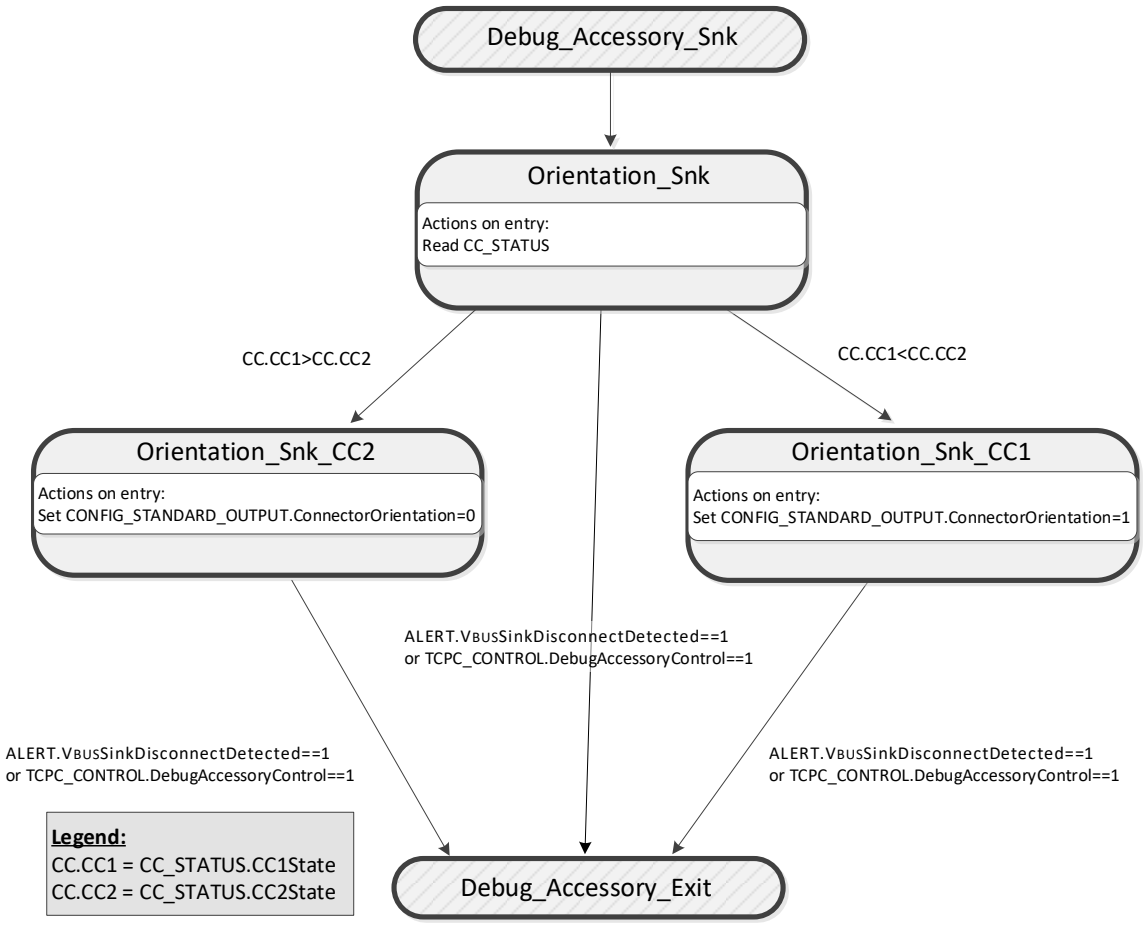


Figure 4-22. TCPC Debug Accessory State-Machine: Sink Orientation

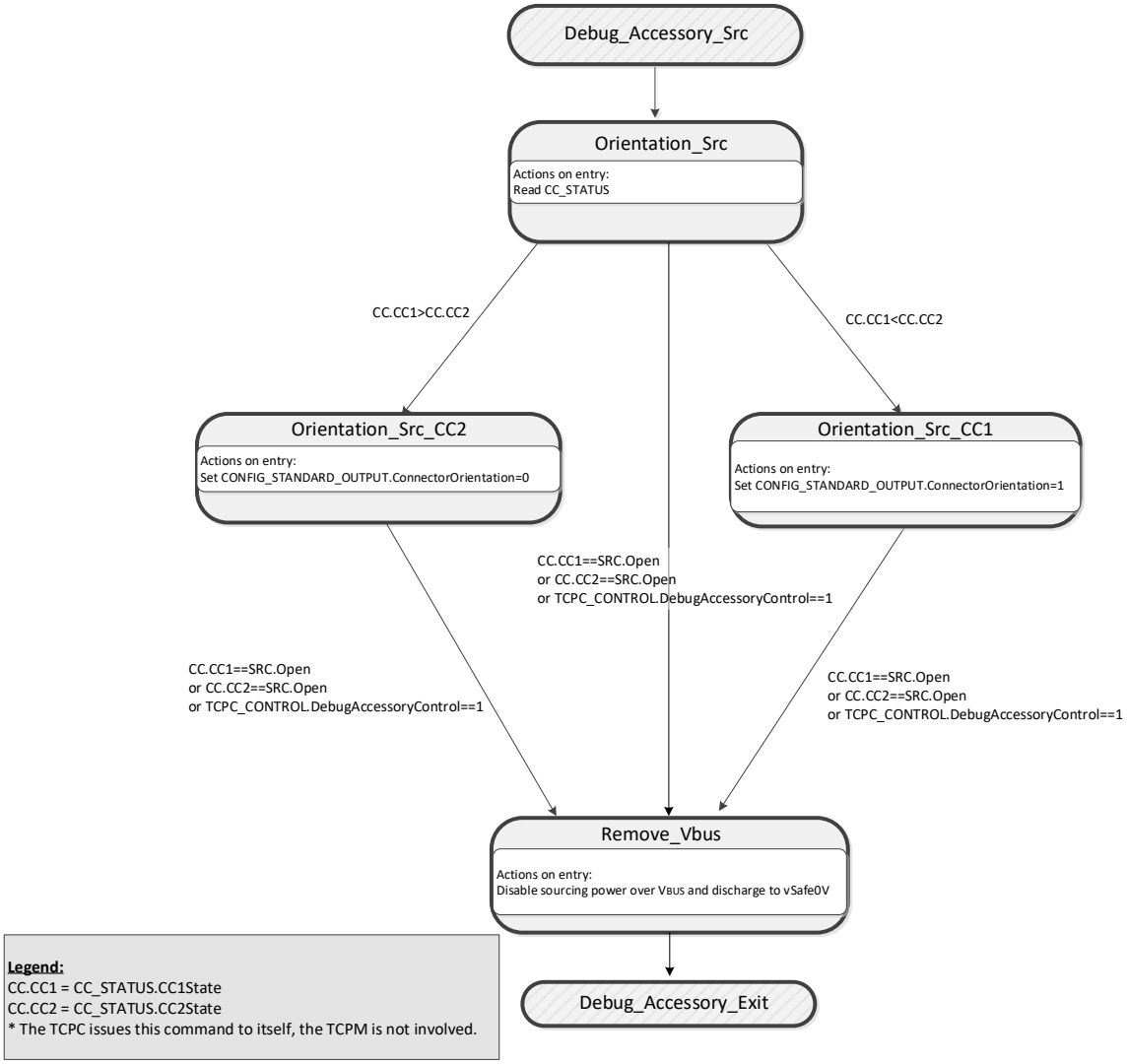


Figure 4-23 TCPC Debug Accessory State-Machine: Source Orientation

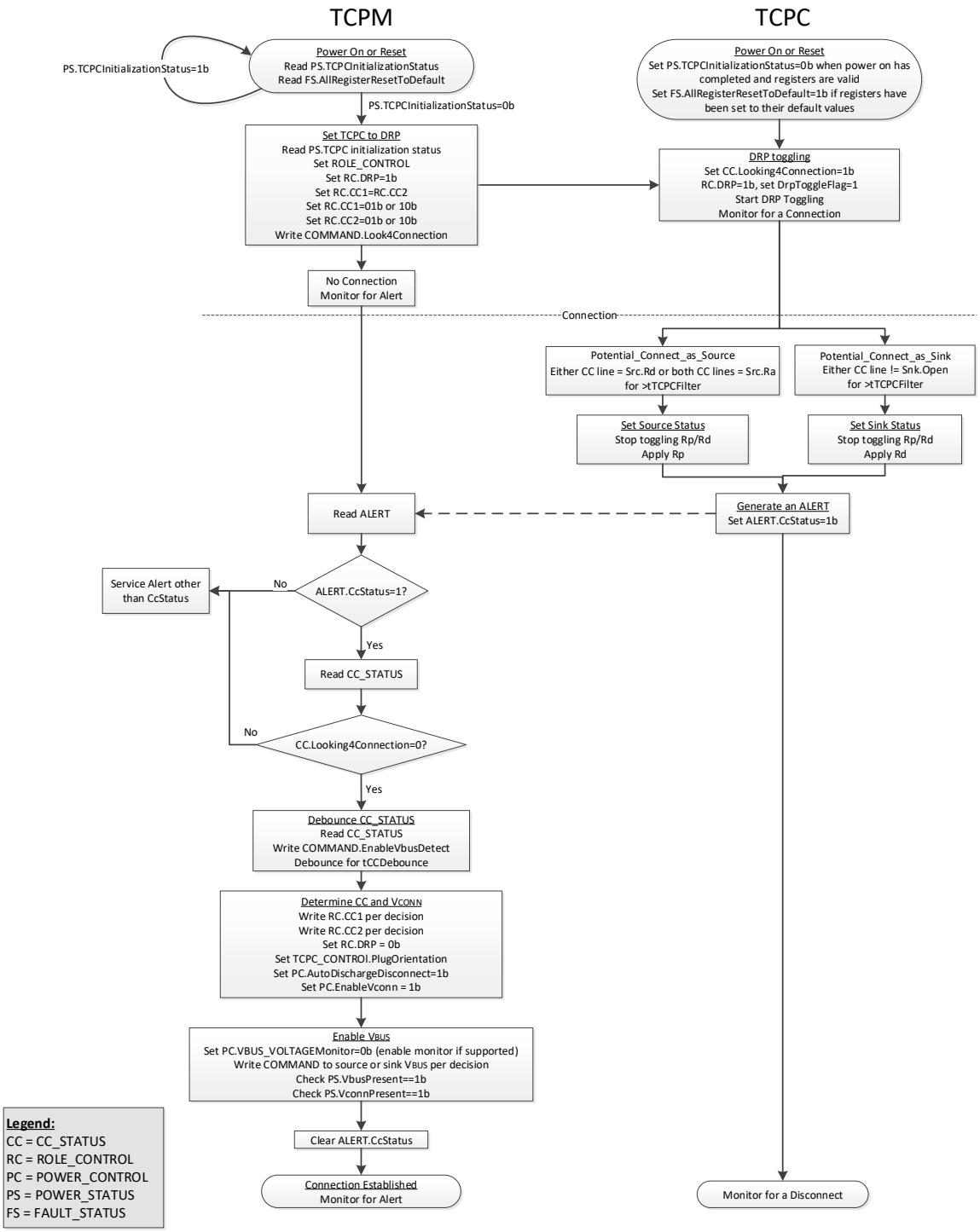


Figure 4-24. DRP Initialization and Connection Detection

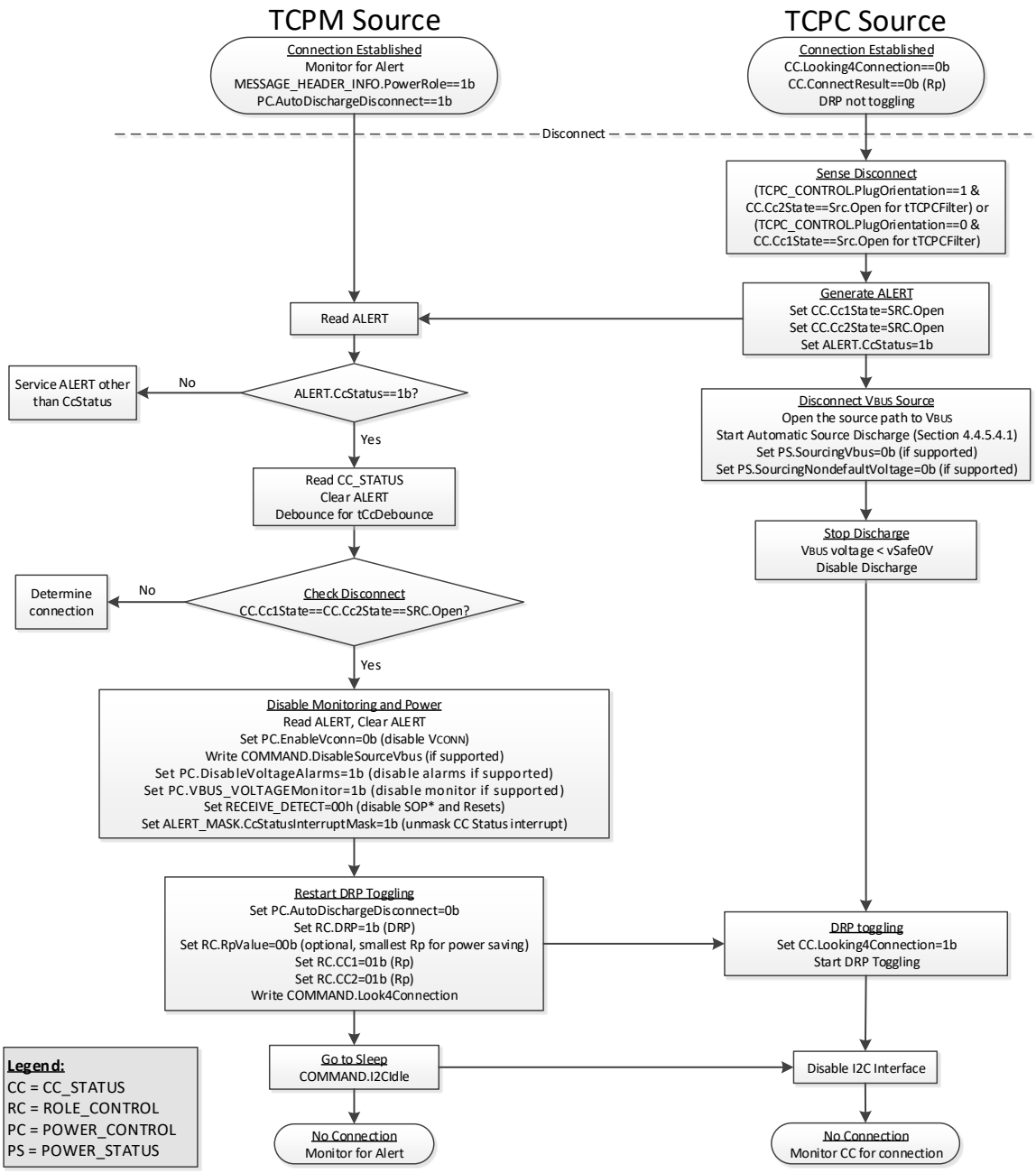


Figure 4-25. Source Disconnect

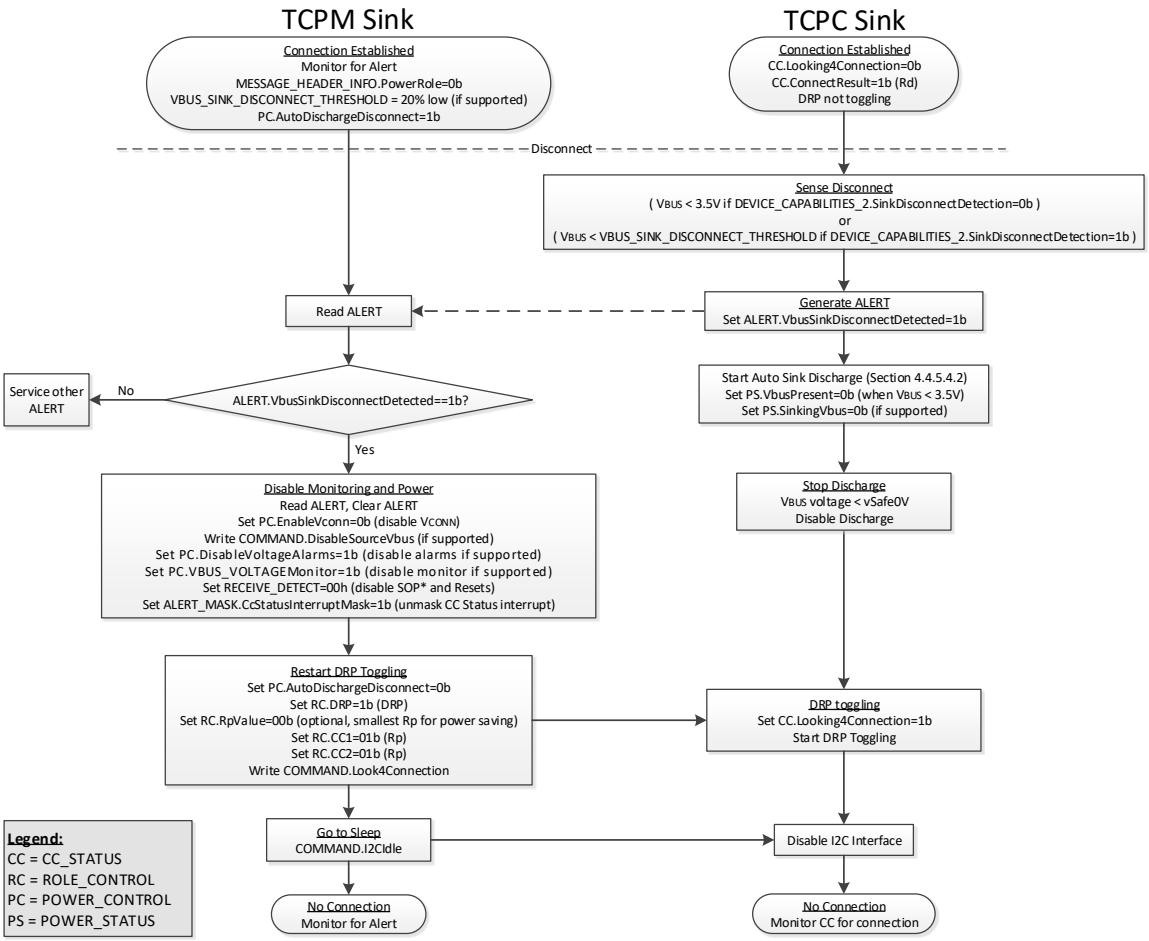


Figure 4-26. Sink Disconnect

4.7 USB PD Communication Operational Model

This section describes the procedures of ***USB PD*** communication through TCPC.

4.7.1 Transmitting an SOP* USB PD Message with Less than or Equal to 128 Data Bytes

The TRANSMIT_BUFFER holds the content of the SOP* ***USB PD*** message to be transmitted. The TCPC automatically increments the TRANSMIT_BUFFER.TX_BUF_BYTE_x offset when the TCPM writes to TRANSMIT_BUFFER.TX_BUF_BYTE_x. For example, after N number of bytes are written to the TRANSMIT_BUFFER.TX_BUF_BYTE_x, the next write to TRANSMIT_BUFFER.TX_BUF_BYTE_x occurs at buffer offset N+1 byte. The TCPM may re-write TRANSMIT_BUFFER.TX_BUF_BYTE_x beginning at offset 1 by writing COMMAND.ResetTransmitBuffer. The steps for transmitting an SOP* ***USB PD*** message are as follows:

1. The TCPM writes the content of the message to be transmitted into the TRANSMIT_BUFFER
2. The TCPM writes to TRANSMIT requesting SOP* transmission.
3. If the TCPM writes to TRANSMIT requesting a transmission that is not Hard Reset, Cable Reset or BIST Carrier Mode 2 (i.e. TRANSMIT.SOP*Message > 100b) and there are less than 2 bytes in the TX_BUF_BYTE_x register (i.e. the transmit buffer pointer is less than offset 3), the TCPC shall generate a FAULT_STATUS.I2CInterfaceError.
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 a 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.
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 an SOP* USB PD Message with Greater than 128 Data Bytes

The TRANSMIT_BUFFER holds the content of the SOP* ***USB PD*** message to be transmitted. If DEVICE_CAPABILITIES_2.LongMessage is set to one, the TRANSMIT _BUFFER is capable of holding 264 byte SOP* messages that include the Message Header, Extended Message Header and the Data. The TCPC automatically increments the TRANSMIT_BUFFER.TX_BUF_BYTE_x offset when the TCPM writes to TRANSMIT_BUFFER.TX_BUF_BYTE_x. For example, after N number of bytes are written to the TRANSMIT_BUFFER.TX_BUF_BYTE_x, the next write to TRANSMIT_BUFFER.TX_BUF_BYTE_x occur at buffer offset N+1 byte. The TCPM may re-write the TRANSMIT_BUFFER.TX_BUF_BYTE_x beginning at offset 1 by writing the COMMAND.ResetTransmitBuffer. If DEVICE_CAPABILITIES_2.LongMessage is set to one, the TCPM may write up to 132 bytes to the TX_BUF_BYTE_x in one burst. Otherwise (DEVICE_CAPABILITIES_2.LongMessage is set to zero), a TCPM may write only up to 30 bytes to the TX_BUF_BYTE_x. If the TCPM writes more than 132 bytes into the TX_BUF_BYTE_x (when DEVICE_CAPABILITIES_2.LongMessage = 1b), the first 132 bytes are written but the remaining overflow bytes are discarded.

The TRANSMIT_BUFFER pointer is reset either when the TCPM writes to the TRANSMIT register or when the TCPM writes COMMAND.ResetTransmitBuffer (0xDD). A TCPC receiving a ***USB PD*** message shall not reset the TRANSMIT_BUFFER. If the TCPM writes to the TRANSMIT register when the TRANSMIT_BUFFER pointer is reset (i.e. the transmit buffer pointer is less offset 3), the TCPC shall generate FAULT_STATUS.I2CInterfaceError.

The TCPM may write TCPC_CONTROL.EnableSMBusPEC = 1 to enable SMBus PEC. If an incorrect PEC is detected as the TCPM writes to TRANSMIT_BUFFER.TX_BUF_BYTE_x, the TCPC shall not automatically increment the TRANSMIT_BUFFER.TX_BUF_BYTE_x offset and shall generate a bit-level NAK to the PEC byte.
The steps for transmitting an SOP* ***USB PD*** message are as follows:

1. The TCPM writes COMMAND.ResetTransmitBuffer (0xDD) to reset the pointer of TRANSMIT_BUFFER to the beginning. This is only necessary if the TRANSMIT register has not been written and the contents of the TRANSMIT_BUFFER.TX_BUF_BYTE_x was previously written.
2. The TCPM writes the TRANSMIT_BUFFER.I2C_WRITE_BYTE_COUNT (N1) and the first portion of the message to be transmitted into TRANSMIT_BUFFER.TX_BUF_BYTE_x.
3. The TCPM writes the TRANSMIT_BUFFER.I2C_WRITE_BYTE_COUNT (N2) and the second portion of the message to be transmitted into TRANSMIT_BUFFER.TX_BUF_BYTE_x. The TCPC inserts these contents into its internal transmit buffer starting at offset N1+1.
   The TCPM writes the TRANSMIT_BUFFER.I2C_WRITE_BYTE_COUNT (N3) and the third portion of the message to be transmitted into TRANSMIT_BUFFER.TX_BUF_BYTE_x. The TCPC inserts these bytes into its internal transmit buffer starting at offset N1+N2+1.
4. The TCPM may repeat Step3 until it has written the entire message.
5. TCPM writes to the TRANSMIT register to request transmitting an SOP* ***USB PD*** message with N1+N2+N3 byte count. By writing to the TRANSMIT register, the pointer of the TRANSMIT_BUFFER is reset. The TCPM must guarantee the bytes are written into the TRANSMIT_BUFFER before requesting the TCPC to place the ***USB PD*** message on the CC wire.
6. The outcome of the TRANSMIT register 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. The pointer of TRANSMIT_BUFFER is reset.
   - If the TCPC PHY layer did not get any response after retries, the TCPC sets the TransmitSOP*MessageFailed bit in the ALERT register. The pointer of TRANSMIT_BUFFER is reset.
   - If the transmission was discarded due to an incoming received message, the TCPC sets the TransmitSOP*MessageDiscarded bit in the ALERT register. The pointer of TRANSMIT_BUFFER is reset.
7. Before requesting another transmission, the TCPM clears the alert by writing a logical 1 to the asserted bit(s) in the ALERT register.

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

Example #1: Successful transmission of 260 data bytes in an Extended ***USB PD*** Message

1. The TCPM intends to transmit 260 data bytes in an Extended ***USB PD*** Message
2. The TCPM writes 133 total bytes: I2C_WRITE_BYTE_COUNT (132) + TX_BUF_BYTE_0…TX_BUF_BYTE_131 (4 header bytes + the beginning 128 data bytes)
3. The TCPC moves the pointer of TRANSMIT_BUFFER to offset 133.
4. The TCPM writes 133 total bytes: I2C_WRITE_BYTE_COUNT (132) + TX_BUF_BYTE_0…TX_BUF_BYTE_131 (the remaining 132 data bytes).
5. The TCPM writes to TRANSMIT register requesting SOP* transmission of 4 header bytes + 260 data bytes.
6. The TCPC PHY layer successfully transmits the 260 data byte ***USB PD*** message, the TCPC sets the TransmitSOP*MessageSuccessful bit in the ALERT register. The pointer of TRANSMIT_BUFFER is reset.

7. The TCPM clears ALERT.TransmitSOP*MessageSuccessful

Example #2: Abbreviated transaction. Successful transmission of 28 data bytes in a ***USB PD*** Message.

1. The TCPM intends to transmit 28 data bytes in a ***USB PD*** Message
2. The TCPM writes 17 total bytes: I2C_WRITE_BYTE_COUNT (30) + TX_BUF_BYTE_0…TX_BUF_BYTE_15 (2 header bytes + 14 data bytes). The TCPM issues a Stop bit to abort the write transaction. The I2C write is ignored and the pointer of TRANSMIT_BUFFER remains at offset 1. The TCPC shall assert FAULT_STATUS.I2CInterfaceError.
3. The TCPM intends to rewrite to transmit the previous 27 data bytes in a ***USB PD*** Message
4. The TCPM writes 31 total bytes: I2C_WRITE_BYTE_COUNT (30) + TX_BUF_BYTE_0…TX_BUF_BYTE_29 (2 header bytes + 28 data bytes).
5. The TCPC moves the pointer of TRANSMIT_BUFFER to offset 31
6. The TCPM writes to TRANSMIT register requesting SOP* transmission of 2 header bytes + 28 data bytes.
7. The TCPC PHY layer successfully transmits the 28 data byte ***USB PD*** message, the TCPC sets the TransmitSOP*MessageSuccessful bit in the ALERT register. The pointer of TRANSMIT_BUFFER is reset.
8. The TCPM clears ALERT.TransmitSOP*MessageSuccessful

Example #3: Using the SMBus PEC, successfully transmitting 260 data bytes in an Extended ***USB PD*** Message. As part of the initialization processes, the TCPM reads DEVICE_CAPABILITIES_2.SMBusPEC = 1 to determine if the TCPC supports SMBus PEC, then TCPM writes TCPC_CONTROL.EnableSMBusPEC = 1 to enable the SMBus PEC. The TCPM may need to limit total number of byte (I2C_WRITE_BYTE_COUNT), to account for SMBus PEC CRC-8 error detection limitation.

1. The TCPM intends to transmit 260 data bytes in an Extended ***USB PD*** Message
2. The TCPM writes 134 total bytes: I2C_WRITE_BYTE_COUNT (132) + TX_BUF_BYTE_0…TX_BUF_BYTE_131 (4 header bytes + the beginning 128 data bytes) + 1 PEC byte. The TCPC checks the validity of the PEC in real time. The TCPC discovers an incorrect PEC and generates a bit-level NAK to the PEC byte. The pointer of TRANSMIT_BUFFER remains at offset 1.
3. The TCPM intends to rewrite to transmit the previous 260 data bytes Extended ***USB PD*** Message
4. The TCPM writes 134 total bytes: I2C_WRITE_BYTE_COUNT (132) + TX_BUF_BYTE_0…TX_BUF_BYTE_131 (4 header bytes + the beginning 128 data bytes) + 1 PEC byte. The TCPC checks the validity of the PEC in real time. The CRC calculated by TCPC matches the PEC byte and it generates a bit-level ACK to the PEC byte.
5. The TCPC moves the pointer of TRANSMIT_BUFFER to offset 133.
6. The TCPM writes 134 total bytes: I2C_WRITE_BYTE_COUNT (132) + TX_BUF_BYTE_0…TX_BUF_BYTE_131 (the remaining 132 data bytes) + 1 PEC byte. The TCPC checks the validity of the PEC in real time. The TCPC discovers an incorrect PEC and generates a bit-level NAK to the PEC byte. The pointer of TRANSMIT_BUFFER remains at offset 133.
7. The TCPM intends to rewrite to the previous remaining 132 data bytes.
8. The TCPM writes 134 total bytes: I2C_WRITE_BYTE_COUNT (132) + TX_BUF_BYTE_0…TX_BUF_BYTE_131 (the remaining 132 data bytes) + 1 PEC byte. The TCPC checks the validity of the PEC in real time. The CRC calculated by TCPC matches the PEC byte and it generates a bit-level ACK to the PEC byte.
9. The TCPM writes to the TRANSMIT register requesting an SOP* transmission of 4 header bytes + 260 data bytes.

10. The TCPC PHY layer successfully transmits the 260 data byte ***USB PD*** message, the TCPC sets the TransmitSOP*MessageSuccessful bit in the ALERT register. The pointer of the TRANSMIT_BUFFER is reset.
11. The TCPM clears ALERT.TransmitSOP*MessageSuccessful.

4.7.3 Transmitting a Hard Reset Message

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

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.
2. The TCPC asserts both ALERT.TransmitSOP*MessageSuccessful and ALERT.TransmitSOP*MessageFailed regardless of the outcome of the transmission and asserts the Alert# pin.
3. The TCPC clears the RECEIVE_DETECT and READABLE_BYTE_COUNT registers to disable the ***USB PD*** message passing.
4. The TCPC resets the mask registers (ALERT_MASK, POWER_STATUS_MASK, EXTENDED_STATUS_MASK, ALERT_EXTENDED_MASK) per Table 4-1
5. 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.
6. The TCPM writes to the RECEIVE_DETECT register to enable ***USB PD*** message passing.

4.7.4 Transmitting a Cable Reset Message

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

1. The TCPM writes to TRANSMIT to request a Cable Reset transmission,
2. The TCPC asserts both ALERT.TransmitSOP*MessageSuccessful and ALERT.TransmitSOP*MessageFailed regardless of the outcome of the transmission and asserts the Alert# pin.

4.7.5 Receiving SOP* USB PD Messages with Less than or Equal to 128 Data Bytes

The steps for receiving a short SOP* ***USB PD*** message are as follows:

1. The TCPC asserts the Alert# pin to request attention when it receives a Hard Reset, Cable Reset, or has sent the GoodCRC in response to an SOP* ***USB PD*** message from a Port Partner. If an overflow has occurred, the TCPC will have set the ALERT.RxBufferOverflow register, therefore the TCPC will not send 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.
2. The TCPM reads the ALERT register and ALERT.ReceiveSOP*MessageStatus is asserted for notification that a message was received.
3. The TCPM reads the RECEIVE_BUFFER.READABLE_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. RECEIVE_BUFFER.RX_BUF_BYTE_x) as defined in the RECEIVE_BUFFER.READABLE_BYTE_COUNT. Note that RECEIVE_BUFFER.RX_BUF_FRAME_TYPE and RECEIVE_BUFFER.RX_BUF_BYTE_x are “hidden” and these registers can only be accessed by reading at address 30h (refer to Table 4-1).
4. The TCPM clears the Alerts:
    - The ALERT.RxBufferOverflow is cleared when the TCPM writes ALERT.ReceiveSOP*MessageStatus to 1 and ALERT.RxBufferOverflow to 1.

- Writing ALERT.ReceiveSOP*MessageStatus to 1 also clears the receive buffer registers.

5. After the Alert and buffers have been cleared, the TCPC shall put the next received message (if any) into RECEIVE_BUFFER.RX_BUF_BYTE_x. The TCPC shall then update RECEIVE_BUFFER.READABLE_BYTE_COUNT and ALERT registers.
6. If Alert# pin is still asserted, return to Step 2.

4.7.6 Receiving SOP* USB PD Messages with Greater than 128 Data Bytes

If DEVICE_CAPABILITIES_2.LongMessage is set to one, the RECEIVE_BUFFER is sized to hold a 264 bytes SOP* message plus a 30 bytes SOP* message.
The steps for receiving a long SOP* message are as follows:

1. 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 send 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.
2. The TCPM reads the ALERT register and ALERT.BeginningSOP*MessageStatus is asserted for notification that an extended ***USB PD*** Message with more than 128 data bytes was received.
3. The TCPM reads the RECEIVE_BUFFER.READABLE_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. RECEIVE_BUFFER.RX_BUF_BYTE_x) as defined in the RECEIVE_BUFFER.READABLE_BYTE_COUNT. At this point, if TCPM writes COMMAND.ResetReceiveBuffer (0xEE), the pointer of RX_BUF_BYTE_x is reset to offset 1.
4. The TCPM clears the Alerts:
    - Writing ALERT.BeginningSOP*MessageStatus to 1 also clears the receive buffer registers. The TCPM cannot read this portion of the message once the alert has been cleared.
5. After the Alert and buffers have been cleared, the TCPC shall put the next part of the received message (if any) into RECEIVE_BUFFER.RX_BUF_BYTE_x. The TCPC shall then update the RECEIVE_BUFFER, the READABLE_BYTE_COUNT, the RX_BUF_FRAME_TYPE and the ALERT registers. The pointer of RX_BUF_BYTE_x is at 133.
6. The TCPM reads the ALERT register and ALERT.ReceiveSOP*MessageStatus is asserted for notification that a message was received.
7. The TCPM reads the RECEIVE_BUFFER.READABLE_BYTE_COUNT and RECEIVE_BUFFER.RX_BUF_FRAME_TYPE. The TCPM reads as many bytes in the buffer (i.e. RECEIVE_BUFFER.RX_BUF_BYTE_x) as defined in the RECEIVE_BUFFER.READABLE_BYTE_COUNT. At this point, if TCPM writes COMMAND.ResetReceiveBuffer (0xEE), the pointer of RX_BUF_BYTE_x is reset to offset 133.
8. The TCPM clears the Alerts:
    - The ALERT.RxBufferOverflow is cleared when the TCPM writes ALERT.ReceiveSOP*MessageStatus to 1 and ALERT.RxBufferOverflow to 1.
    - Writing ALERT.ReceiveSOP*MessageStatus to 1 also clears the receive buffer registers.
9. After the Alert and buffers have been cleared, the TCPC shall put the next received message (if any) into RECEIVE_BUFFER.RX_BUF_BYTE_x. The TCPC shall then update RECEIVE_BUFFER.READABLE_BYTE_COUNT and ALERT registers.

If the TCPC asserts the Alert# pin (e.g. due to ALERT_EXTENDED.SinkFRSwap being set) while the TCPM is reading the RECEIVE_BUFFER for a long SOP* message, the TCPM may issue a Stop bit to abort the read transaction and service the ALERT register.

Example: First receiving an Extended ***USB PD*** Message with 260 data bytes followed by receiving a second ***USB PD*** Message with 28 data bytes:

1. The TCPC receives an Extended ***USB PD*** message with 260 data bytes. TCPC then receives a second ***USB PD*** message with 28 data bytes before the TCPM reads the first message.
2. After the GoodCRC is sent, the TCPC asserts ALERT.BeginningSOP*MessageStatus
3. The TCPM reads ALERT.
4. The TCPM reads 134 total bytes: READABLE_BYTE_COUNT (133) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_0…RX_BUF_BYTE_131 (4 header bytes + 128 data bytes)
5. TCPM clears ALERT.BeginningSOP*MessageStatus.
6. TCPC clears the beginning of the first ***USB PD*** message in the receive buffer (4 header bytes and the beginning 128 data bytes). Then, TCPC loads the remaining 132 data bytes of the first ***USB PD*** message into the receive buffer registers.
7. TCPC asserts ALERT.ReceiveSOP*MessageStatus
8. TCPM reads ALERT.
9. TCPM reads 134 total bytes: READABLE_BYTE_COUNT (133) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_0…RX_BUF_BYTE_131 (132 data bytes)
10. TCPM clears ALERT.ReceiveSOP*MessageStatus
11. TCPC clears the remaining of the first ***USB PD*** message in the receive buffer (132 data bytes). Then, TCPC loads the second ***USB PD*** message (with 28 data bytes) into the receive buffer registers.
12. TCPC asserts ALERT.ReceiveSOP*MessageStatus
13. TCPM reads ALERT.
14. TCPM reads 32 total bytes: READABLE_BYTE_COUNT (31) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_0…RX_BUF_BYTE_29 (2 header bytes + 28 data bytes)
15. TCPM clears ALERT.ReceiveSOP*MessageStatus

4.7.7 Re-Reading RECEIVE_BUFFER

The TCPC automatically increments the pointer of RECEIVE_BUFFER.RX_BUF_BYTE_x when the TCPM reads RECEIVE_BUFFER.RX_BUF_BYTE_x. For example, after N number of bytes are first read from the RECEIVE_BUFFER.RX_BUF_BYTE_x, the next read of RECEIVE_BUFFER.RX_BUF_BYTE_x starts at buffer offset N+1 byte. However, the pointer of RECEIVE_BUFFER.RX_BUF_BYTE_x is not incremented if the TCPM reads the RECEIVE_BUFFER.READABLE_BYTE_COUNT or the RECEIVE_BUFFER.RX_BUF_FRAME_TYPE. The TCPM may re-read the RECEIVE_BUFFER.RX_BUF_BYTE_x by writing COMMAND.ResetReceiveBuffer (0xEE).

Example #1: Reading the READABLE_BYTE_COUNT and the RX_BUF_FRAME_TYPE separately, and then block read RX_BUF_BYTE_x,

1. The TCPC receives a ***USB PD*** message with 28 data bytes.
2. After the GoodCRC is sent, the TCPC asserts ALERT.ReceiveSOP*MessageStatus
3. The TCPM reads 2 total bytes: READABLE_BYTE_COUNT (31) + RX_BUF_FRAME_TYPE. The pointer of RX_BUF_BYTE_x remains at offset 1.
4. The TCPM reads 32 total bytes: READABLE_BYTE_COUNT (31) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_0…RX_BUF_BYTE_29 (2 header bytes + 28 data bytes)
5. The TCPM clears ALERT.ReceiveSOP*MessageStatus

Example #2: Abbreviated transaction

1. The TCPC receives a ***USB PD*** message with 28 data bytes.
2. After the GoodCRC is sent, the TCPC asserts ALERT.ReceiveSOP*MessageStatus.
3. The TCPM reads ALERT.

4. The TCPM reads 17 total bytes: READABLE_BYTE_COUNT (31) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_0…RX_BUF_BYTE_14 (2 header bytes + 13 data bytes). The TCPM issues a Stop bit to abort the read transaction.
5. The TCPM writes COMMAND.ResetReceiveBuffer (0xEE). The pointer of RX_BUF_BYTE_x is moved to offset 1.
6. The TCPM reads 32 total bytes: READABLE_BYTE_COUNT (31) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_0…RX_BUF_BYTE_29 (2 header bytes + 28 data bytes)
7. The TCPM clears ALERT.ReceiveSOP*MessageStatus

Example #3: Using SMBus PEC. When SMBus PEC is enabled, the TCPM shall read as many bytes in the buffer as defined in the RECEIVE_BUFFER.READABLE_BYTE_COUNT in one I2C read transaction. The TCPM may reread the RECEIVE_BUFFER to account for the error detection limitation of SMBus PEC CRC-8. In the case of repeated reading of the RECEIVE_BUFFER, the TCPM should run the I2C at Fm+ bus speed for optimal flow control.

1. The TCPM reads DEVICE_CAPABILITIES_2.SMBusPEC = 1 to determine the TCPC supports SMBus PEC
2. The TCPM writes TCPC_CONTROL.EnableSMBusPEC = 1 to enable the SMBus PEC
3. The TCPC receives an extended ***USB PD*** message with 260 data bytes.
4. After the GoodCRC is sent, the TCPC asserts ALERT.BeginningSOP*MessageStatus.
5. The TCPM reads 135 total bytes: READABLE_BYTE_COUNT (133) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_0…RX_BUF_BYTE_131 (4 header bytes + 128 data bytes) + 1 PEC byte.
6. The CRC calculated by TCPM does not match the PEC byte.
7. The TCPM writes COMMAND.ResetReceiveBuffer (0xEE). The pointer of RX_BUF_BYTE_x is moved to offset 1.
8. The TCPM rereads 135 total bytes: READABLE_BYTE_COUNT (133) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_0…RX_BUF_BYTE_131 (4 header bytes + 128 data bytes) + 1 PEC byte.
9. The CRC calculated by TCPM matches the PEC byte.
10. The TCPM clears ALERT.BeginningSOP*MessageStatus.
11. The TCPC clears the beginning of the ***USB PD*** message in the receive buffer (4 header bytes and the beginning 128 data bytes). Then, TCPC loads the remaining 132 data bytes into the receive buffer registers. The pointer of RX_BUF_BYTE_x is moved to offset 133.
12. The TCPC asserts ALERT.ReceiveSOP*MessageStatus.
13. The TCPM reads ALERT.
14. The TCPM reads 135 total bytes: READABLE_BYTE_COUNT (133) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_0…RX_BUF_BYTE_131 (132 data bytes) + 1 PEC byte.
15. The CRC calculated by TCPM does not match the PEC byte.
16. The TCPM writes COMMAND.ResetReceiveBuffer (0xEE). The pointer of RX_BUF_BYTE_x is moved to offset 133.
17. The TCPM reads 135 total bytes: READABLE_BYTE_COUNT (133) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_0…RX_BUF_BYTE_131 (132 data bytes) + 1 PEC byte.
18. The CRC calculated by TCPM matches the PEC byte.
19. The TCPM clears ALERT.ReceiveSOP*MessageStatus.
20. The TCPC clears the remaining of the ***USB PD*** message in the receive buffer (132 data bytes).

Example #4: Multiple read transactions of RX_BUF_BYTE_x. The following flow is not SMBus compliant but shall be supported by TCPC when SMBus PEC is disabled (i.e. TCPC_CONTROL.EnableSMBusPEC = 0).

1. The TCPC receives a ***USB PD*** message with 28 data bytes.
2. After the GoodCRC is sent, the TCPC asserts ALERT.ReceiveSOP*MessageStatus.
3. The TCPM reads ALERT.

4. The TCPM reads 17 total bytes: READABLE_BYTE_COUNT (31) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_0…RX_BUF_BYTE_14 (2 header bytes + 13 data bytes). The TCPM issues a Stop bit to terminate the read transaction.
5. The TCPM reads 17 total bytes: READABLE_BYTE_COUNT (16) + RX_BUF_FRAME_TYPE + RX_BUF_BYTE_15…RX_BUF_BYTE_29 (15 data bytes).
6. The TCPM clears ALERT.ReceiveSOP*MessageStatus.

4.7.8 Receiving a Hard Reset message

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

1. The TCPC asserts Alert# pin to request attention when it receives a Hard Reset
2. The TCPC shall set the RECEIVE_DETECT bits to zero to disable automatic transmission of GoodCRC.
3. The TCPC shall set the RECEIVE_BUFFER.READABLE_BYTE_COUNT to zero.
4. The TCPC shall reset the mask registers (ALERT_MASK, POWER_STATUS_MASK, EXTENDED_STATUS_MASK, ALERT_EXTENDED_MASK) per Table 4-1

4.7.9 Receiving a Cable Reset message

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

1. 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.
2. The TCPC shall set the RECEIVE_DETECT bits to zero to disable automatic transmission of GoodCRC.
3. The TCPC shall set the RECEIVE_BUFFER.READABLE_BYTE_COUNT to one.

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

The TCPM may put the I2C device interface in an idle state by issuing COMMAND.I2Cidle. The TCPC may then generate a bit-level NAK 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 shall not be NAK'd. The steps for the TCPM to wake a TCPC from I2C idle:

1. TCPM sends an I2C command to address the TCPC (or COMMAND.WakeI2C)
2. 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).
3. The TCPC updates the ALERT.PowerStatus bit and then sets the ALERT#. The TCPC does not set any registers in the POWER_STATUS register.
4. The TCPM stops the ***tWakeI2Cfail*** timer.
5. TCPM reads ALERT.PowerStatus registers to get notification the TCPC has exited I2C idle
6. 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

4.8.2 USB PD Message Delivery

The TCPM may disable ***USB PD*** message delivery by setting the RECEIVE_DETECT and READABLE_BYTE_COUNT registers to all zeroes. Disabling ***USB PD*** message delivery 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 the 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 Detection

The TCPC shall disable VBUS present and vSafe0V detection circuits when the TCPM issues COMMAND.DisableVBUSDetect. When the VBUS detection is disabled:

- TCPC shall set POWER_STATUS.VbusDetectionEnabled = 0b
- TCPM shall ignore POWER_STATUS.VBUSPresent bit
- TCPM shall ignore EXTENDED_STATUS.vSafe0V bit
- Issuing COMMAND.DisableVBUSDetect does not cause ALERT register 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 the 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 setting POWER_CONTROL.AutoDischargeDisconnect to one.
For a Sink, Auto Discharge is disabled by either setting all zeros to VBUS_SINK_DISCONNECT_THRESHOLD or setting POWER_CONTROL.AutoDischargeDisconnect to 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 the FAULT_CONTROL register to all 1s.
When one of the bits in FAULT_CONTROL is set to 1 to disable the fault status reporting:

- The TCPC shall set the corresponding bit in the FAULT_STATUS register to zero and TCPM shall ignore that bit.

4.9 Type-C Port Controller Timing Constraints

The TCPC shall comply with the timing constraints defined in Table 4-49. The considerations for the TCPM can be found in Appendix B.

Table 4-49. TCPC Timing Constraints

Symbol	Parameter	Min	Max	Units
tBUFFER2CC	Time between I2C STOP and the first bit of the Preamble		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
tTCPCSendFRSwap	Time between receiving request to send Fast Role Swap signal and sending the signal		50	us

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 4-50 and Table 4-51.
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 (the 1st time 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.

Table 4-50. I2C Static Characteristics

Symbol	Parameter	Conditions	Min	Max	Units
VDD	I/O Supply Voltage		1.8	3.6	Volts
VIL	LOW-level input voltage		-0.5	0.3VDD	Volts
VIH	HIGH-level input voltage		0.7VDD	VDD+0.5	Volts
II	Input current each IO Pin	0.1VDD < VIL OR VIH < 0.9VDDMAX	-10[3]	+10[3]	uA

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

Table 4-51. 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).
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 or the TCPM writes to a register/bit that is not implemented/reserved.

A Informative TCPM Protocol Layer State Diagrams

This section provides informative TCPM Protocol Layer state diagrams to aid the TCPM designer. The 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 PD*** if receiving an unexpected GoodCRC message.

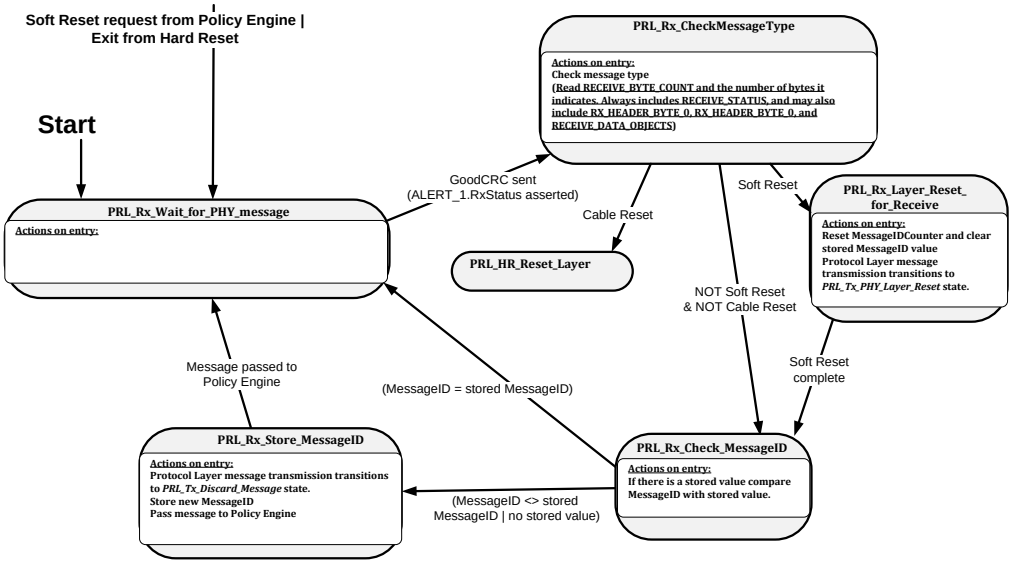


Figure A-1. Message Reception State Diagram Implemented in TCPM

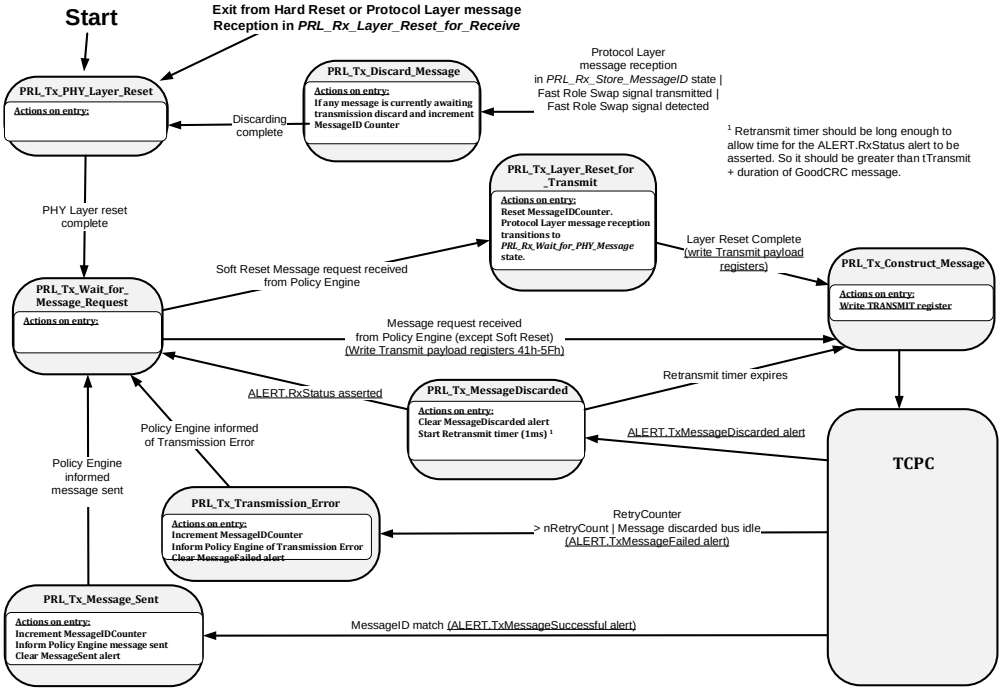


Figure A-2. Message Transmission State Diagram Implemented in TCPM

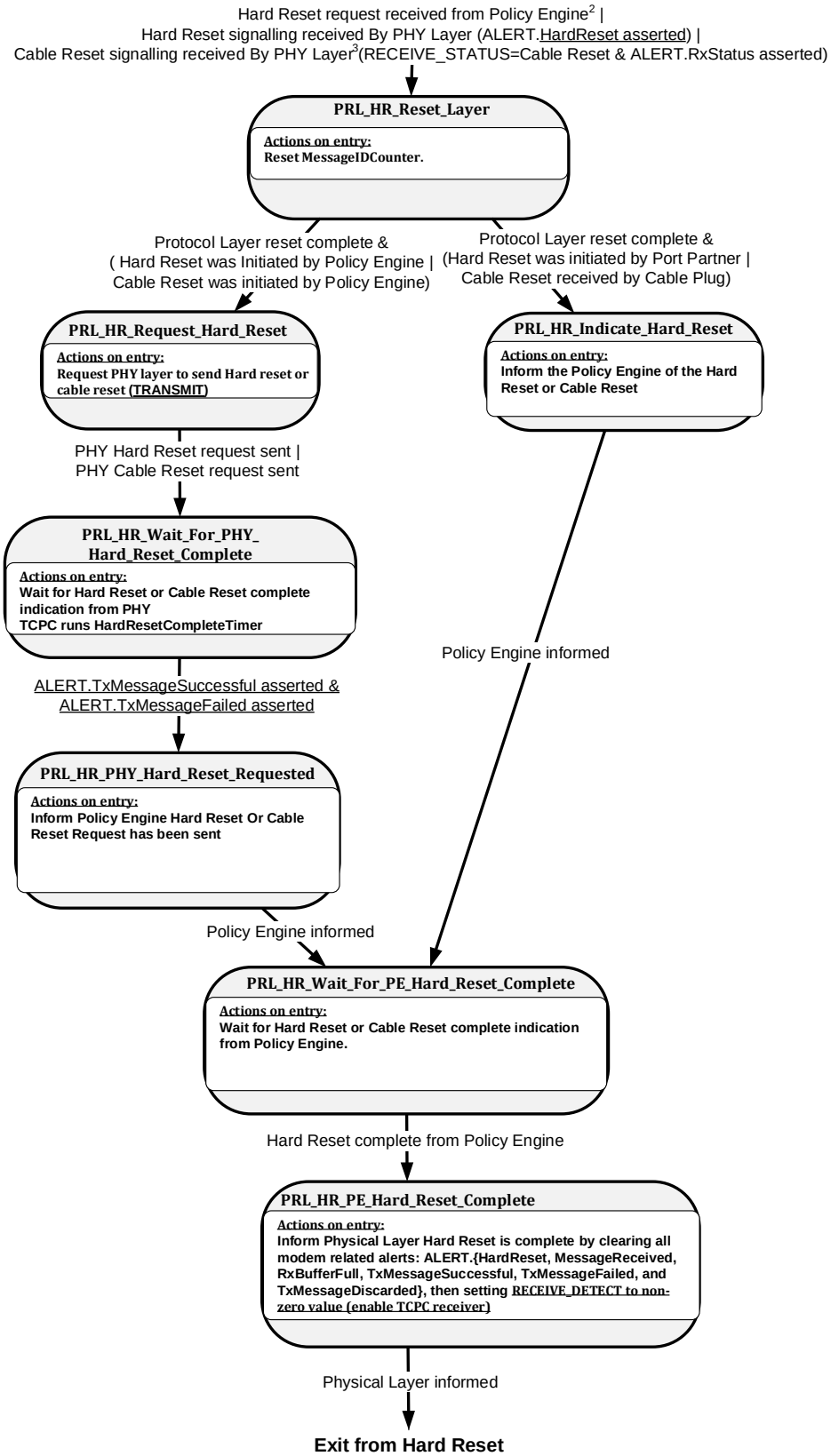


Figure A-3. Hard Reset State Diagram Implemented in TCPM

B Informative TCPM Timing Considerations

This section outlines considerations for a TCPM to TCPC interface. The number of TCPCs which may be supported on one I2C interface from the TCPC depends upon the I2C bus speed and the USB PD target application (e.g. whether the TCPM is a USB Type-C Authentication Initiator, Authentication Responder, or it supports USB-PD Firmware Update). A TCPM can allocate more than one I2C buses to service the TCPCs to overcome the limitation of number of USB Type-C port supported.
The following table summarizes the number of TCPCs on one I2C bus for a given TCPM implementation considering the USB PD application. If the TCPM implements multi-threaded real-time OS, more TCPC ports may be supported on one I2C interface.

Table 4-52. Implementations and Impact on TCPCs on one I2C Interface

TCPM Implementation Assumptions[1]	I2C Requirement		USB PD Application			
	Max number of TCPCs on one I2C	*I2C Bus Speed*	*USB-PD with 7 data objects*	*USB Type-C Authentication Initiator*	*USB Type-C Authentication Responder*	*USB-PD Firmware Update*
5ms or less to enter TCPM task and start servicing ***USB PD*** messages. 1ms or less to respond to ***USB PD*** message while servicing upper layer interrupts and maintaining states. Same as TCPCI Specification Rev1.0 v1.2	2	400KHz	Yes	Yes[2]	No	Yes[2]
	4	1MHz	Yes	Yes[2]	No	Yes[2]
1ms or less to enter TCPM task and start servicing ***USB PD*** messages. 0.1ms or less to respond to a ***USB PD*** message while servicing upper layer interrupts and maintaining states.	1	400KHz	Yes	Yes[2]	Yes	Yes[2]
	3	1MHz	Yes	Yes[2]	Yes	Yes[2]

Notes
1. Multi-threaded real-time OS in TCPM is not assumed
2. The TCPM owns the ***USB PD*** data flow control. The TCPM should abort reading a long message when other TCPC asserts the Alert# pin.

In a scenario where all ports receive an SOP* message with 30 bytes content simultaneously, and each port is required to respond with an SOP* message with 30 bytes content 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 per Table 4-49 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 ***USB PD*** Messages of the 4 ports
- The host requires 1ms or less to respond to a ***USB PD*** Message while servicing upper layer interrupts and maintaining states.

In a scenario where all ports receive an SOP* message with 10 bytes content simultaneously (such as the GET_CERTIFICATE Authentication Request), and each port is required to respond with an SOP* message with 264 bytes content within 20ms (the upper bound of tCertSent) on each port, the actual value of the timing parameters below affects the number

of ports that can be supported. If timing constraints per Table 4-49 are met, the TCPM may support up to 3 ports with the following assumptions:

- The host operates the I2C at F_{SCL} = 1MHz
- The host requires 1ms or less to enter TCPM task and start servicing ***USB PD*** Messages of the 3 ports
- The host requires 0.1ms or less to respond to a ***USB PD*** Message while servicing upper layer interrupts and maintaining states.

C TCPM Timing Constraints when Acting as a Source and Setting SinkTxOK

The TCPM shall meet the timing requirement ***tSinkTxOKService*** for servicing TCPC Alerts if the TCPM is connected to a TCPC Source and has set SinkTxOK to indicate to the Sink it is OK to send Atomic Message Sequences (AMS).
The ***USB PD*** specification sets the timing requirements when the TCPM is not connected to a TCPC acting as a Source or if the TCPC Source has not set SinkTxNG on its TCPC to indicate the Sink can send Atomic Message Sequences.
Servicing the TCPC Alert includes clearing the Alert and reading/clearing the RECEIVE_BUFFER if needed.

Table 4-53. TCPC Alert Servicing Timing

Symbol	Parameter	Typ	Max	Units
tSinkTxOKService	Time to service the TCPC Alert and read/clear the RECEIVE_BUFFER if needed	5	15	ms