TPS6599xAD Host Interface Technical Reference Manual

Technical Reference Manual

Literature Number: SLVUBT3

May 2020

Contents

List of Figures

List of Tables

Chapter 1
SLVUBT3–May 2020

Introduction

1.1 Introduction

1.1.1 Purpose and Scope

This document describes the Host Interface for the TPS65994AD, TPS65993AD, TPS65992SAD, and TPS65992DAD Type-C Port Switch / Power Delivery (PD) Controller devices. This document is aligned with patch F509.04.02 for TPS65993AD and TPS65994AD. This document is also aligned with patch F509.05.02 for TPS65992SAD and TPS65992DAD.

1.1.2 Related Documents

- Universal Serial Bus Specification, Revision 2.0, April 27, 2000 plus ECN and Errata. http://www.usb.org/developers/docs/usb20_docs/
- Battery Charging Specification, Revision 1.2, December 7, 2010 plus Errata.
- Universal Serial Bus 3.1 Specification, Revision 1.0, July 26, 2013 and ECNs approved through August 11, 2014. www.usb.org/developers/docs
- USB Power Delivery Specification Revision 3.0, Version 1.2, June 21, 2018 www.usb.org/developers/docs
- USB Type-C Cable and Connector Specification Revision 1.3, July 14, 2017. www.usb.org/developers/docs
- VESA DisplayPort (DP) Standard, Version 1.3, September 17, 2014.
- Proposed DisplayPort Alt Mode on USB Type-C Standard, Version 1, Draft 5, September 6, 2014.

1.2 PD Controller Host Interface Description

1.2.1 Overview

The PD Controller provides two physical I2C slaves. The I2C_EC slave is meant to be connected to an Embedded Controller (EC). The I2C2s slave is meant to be connected to a secondary controller such as a Thunderbolt controller. For dual-port PD controllers, both of the I2C slaves respond to two slave addresses; one for each Type-C port. Figure 1-1 shows the logical connections for the system for a dual-port PD controller.

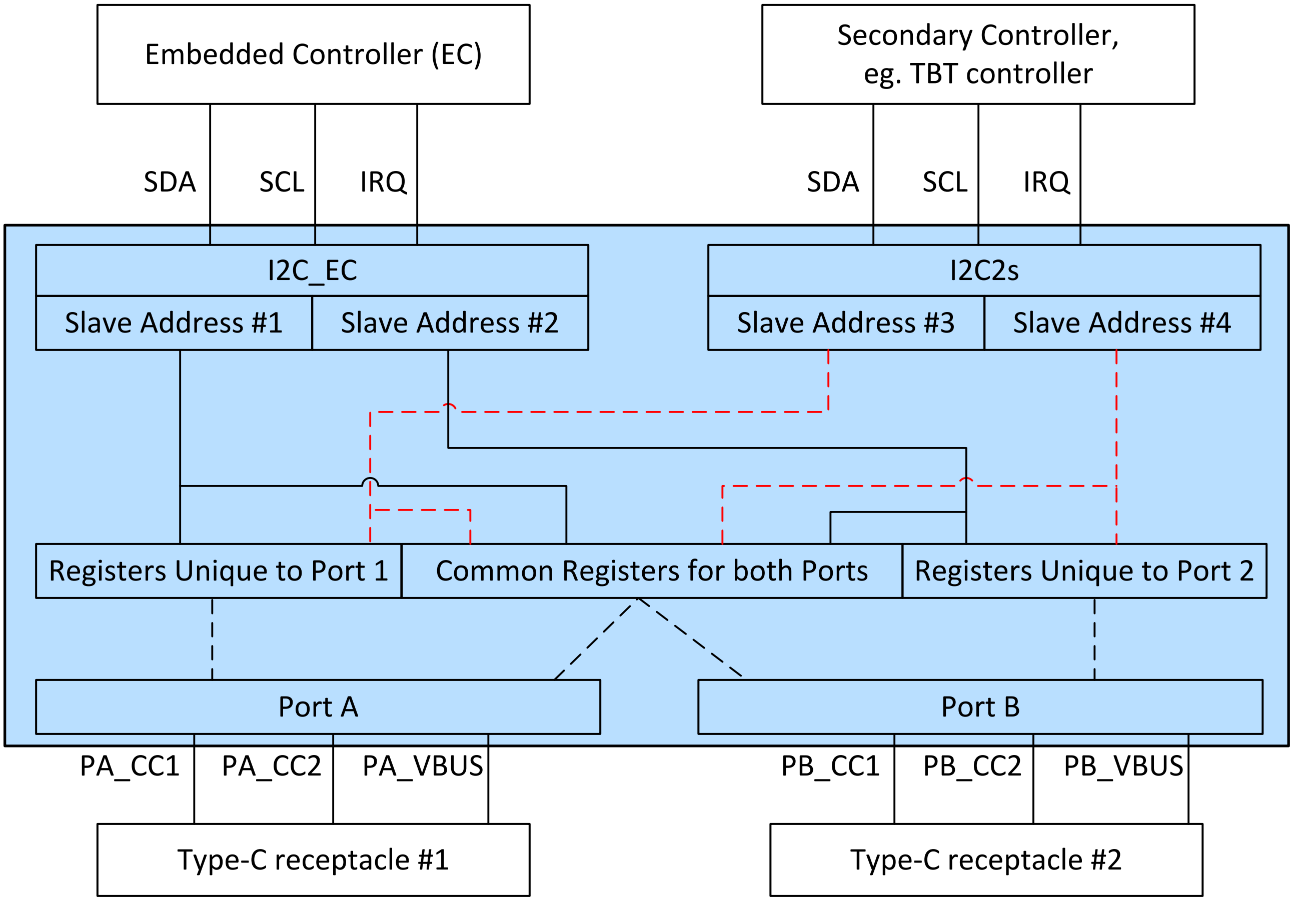


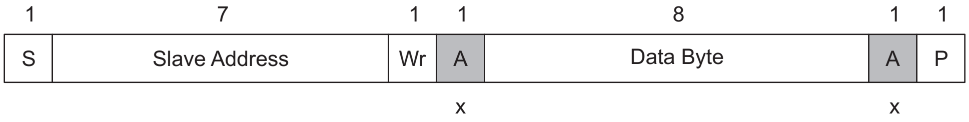
Figure 1-1. Logical connections to registers and Type-C ports in a dual-port PD controller.

The Host Interface defines how the registers are accessed from all I2C slave ports and all slave addresses. Slave Address #1 and Slave Address #2 for I2C_EC are selected by the customer using the ADCIN1 and ADCIN2 pins on the PD controller. Slave Address #3 and Slave Address #4 for I2C2s can be selected throught the GLOBAL_SYSTEM_CONFIG register (0x27). See also Table 5-1 for more details about the slave addresses.

The Host Interface provides general status information to the master of these I2C interfaces about the PD Controller, ability to control the PD Controller, status of USB Type-C Port and communications to/from a connected device (Port Partner) and/or cable plug via USB PD messages. All Host Interface communication that uses the Unique I2C address is referred to as Unique Address Interface.

The PD Controller supports a register-based Unique Address Interface. Section 1.3.2 lists the Unique Address Interface registers and Chapter 3 provides detailed Unique Address Interface register descriptions. For dual-port PD controllers, some registers are not unique per port; so the PD controller only has one instantiation. Other registers are unique per port, and can be accessed from either I2C slave using the slave address associated with that port.

The key to the protocol diagrams is in the SMBus Specification, version 2.0 and is repeated here in part in Figure 1-2.



S	Start Condtion
Sr	Repeated Start Condition
Rd	Read (bit value of 1)
Wr	Write (bit value of 0)
x	Field is required to have the value ‘x’
A	Acknowledge (this bit position may be ‘0’ for an ACK or ‘1’ for a NACK)
P	Stop Condition
□	Master-to-Slave
▨	Slave-to-Master

Figure 1-2. I2C Read/Write Protocol Key

1.2.2 *Register and field notation*

In this document the register names use an ALL CAPS notation, and the field names use a CamelBack notation. For example *TX_SOURCE_CAPS* refers to register 0x32, and *TX_SOURCE_CAPS.numValidPDOs* refers to a specific field in Byte 1 of that register.

Some registers have the same definition, but there are multiple instantiations at different register addresses. The following lists these registers.

- *INT_EVENTx* ~ *INT_EVENT1* (0x14) or *INT_EVENT2* (0x15)
- *INT_MASKx* ~ *INT_MASK1* (0x16) or *INT_MASK2* (0x17)
- *INT_CLEARx* ~ *INT_CLEAR1* (0x18) or *INT_CLEAR2* (0x19)
- *CMDx* ~ *CMD1* (0x08) or *CMD2* (0x10)
- *DATAx* ~ *DATA1* (0x09) or *DATA2* (0x11)

In this document, the "x" indicates that it is referring to both instantiations of that register.

1.3 Unique Address Interface

1.3.1 *Unique Address Interface Protocol*

The Unique Address Interface allows for complex interactions between an I2C master and a single PD Controller. The I2C Slave unique address is used to receive or respond to Host Interface protocol commands. Figure 1-3 and Figure 1-4 show the write and read protocols, respectively. The Byte Count used during a register write may be longer than the number of bytes actually written, in other words the master may issue the stop bit without writing N bytes. Similarly, during a register read, the master may issue the stop bit before reading all N bytes.

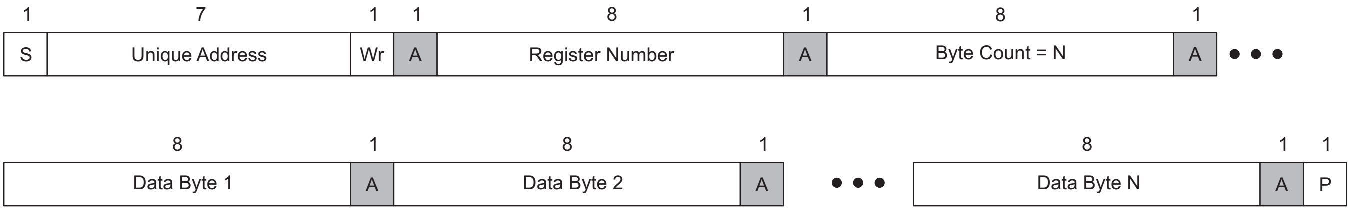


Figure 1-3. I2C Unique Address write register protocol

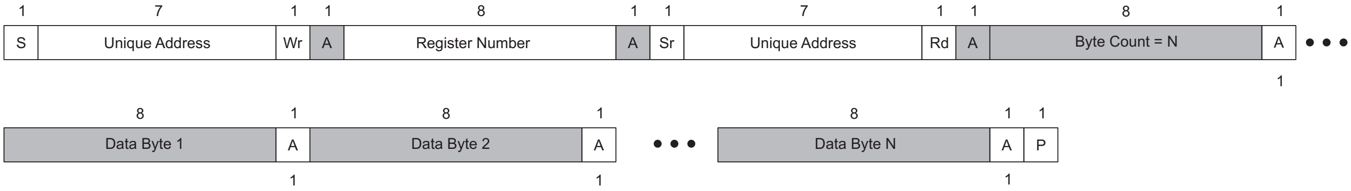


Figure 1-4. I2C Unique Address read register protocol

1.3.2 Unique Address Interface Registers

The PD Controller supports Unique Address Interface registers (Unique Address Registers) provided in Table 1-1. Unless otherwise indicated, 2 or 4 byte registers are little endian (least significant byte in Data Byte 1). Registers that use four character codes (4CC) are defined where the first character corresponds to the ASCII value of Data Byte 1, the second character corresponds to the ASCII value of Data Byte 2, and so forth. Any 4CC codes that are less than 4 characters pad the tail with spaces (0x20).

From a user perspective, the two ports contained in a dual-port PD controller behave as if there are two separate devices in the system that share I2C_EC and I2C2s interfaces (see Figure 1-1) and all registers are logically duplicated. In the following table, the Unique column indicates which registers are logically duplicated for each port (this is only applicable for a dual-port PD controller). For registers that are marked as not unique, the host may read that register using either slave address and it will read back the same value. For registers that are marked as not unique, the host may write that register using either slave address and it will affect both ports. For registers marked as Unique, the registers are logically duplicated for each port and operate independently.

In the register map below, only the registers addresses shown are implemented. All other register addresses are Read-Only Reserved registers that should be ignored.

Table 1-1. Unique Address Interface Registers

Register Number[(1)]	Register Name	Access	# Data Bytes	Unique per Port	Description
0x00	VID	RO	4	no	Intel-assigned Thunderbolt Vendor ID, with the most significant 8 bits of the field padded with 0's. OTP boot loader will use TI's Vendor ID by default. This register may be changed during application customization.
0x01	DID	RO	4	no	Vendor-specific Device ID. Boot loader will use Device ID specific to part (expected to be different per TI part number). This register may be changed during application customization.
0x02	PROTOVER	RO	4	no	Thunderbolt Protocol Version. Required to return 1 per current specification.
0x03	MODE	RO	4	no	Indicates the operational state of the port. The PD controller has limited functionality in some modes. See Section 3.1
0x04	TYPE	RO	4	no	Default response is 'I2C ' (note space as 4th character).
0x05	UID	RO	16	yes	128-bit unique ID (unique for each PD Controller Port)
0x06	CUSTUSE	RO	8	yes	These 8 bytes are allocated for customer use as needed. The PD controller does not use this register. This register may be changed during application customization.
0x07	Reserved				
0x08	CMD1	RW	4	yes	Command register for the primary command interface. Cleared to 0x0000_0000 by the PD Controller during initialization and after successful processing of every command. If an unrecognized command is written to this register, it is replaced by a 4CC value of "!CMD".
0x09	DATA1	RW	64	yes	Data register for the primary command interface (CMD1).
0x0A-0x0C	Reserved				
0x0D	DEVICE_CAPABILITIES	RO	4	no	Description of supported features. See Section 3.2
0x0E	Reserved				

[(1)] Any register number not shown is reserved for proprietary use.

Table 1-1. Unique Address Interface Registers (continued)

Register Number[(1)]	Register Name	Access	# Data Bytes	Unique per Port	Description
0x0F	VERSION	RO	4	no	Binary Coded Decimal version number, bootloader/application code version. Represented as VVVV.MM.RR with leading 0's removed.e.g. 65794d (decimal) to 0x00010102 to 0001.01.02 to 1.1.2 (version). The version information is returned in little Endian format i.e. byte 1 = RR, byte 2 = MM, etc. The 16-bit field RR is used as the FW version in the Source Capabilities Extended and Sink Capabilities Extended messages.
0x10	CMD2	RW	4	yes	Command register for the secondary command interface. Shall be cleared to 0x00000000 by PD Controller during initialization and after successful processing of every command. If an unrecognized command is written to this register it shall be replaced by a 4CC value of "!CMD".
0x11	DATA2	RW	64	yes	Data register used for the secondary command interface (CMD2).
0x12	Reserved				
0x13	Reserved				
0x14	INT_EVENT1	RO	11	yes	Interrupt event bit field for I2C_EC_IRQ. If any bit in this register is 1, then the I2C_EC_IRQ pin is pulled low. See Section 3.3
0x15	INT_EVENT2	RO	11	yes	Interrupt event bit field for I2C2s_IRQ. If any bit in this register is 1, then the I2C2s_IRQ pin is pulled low. See Section 3.3
0x16	INT_MASK1	RW	11	yes	Interrupt mask bit field for INT_EVENT1. A bit in INT_EVENT1 cannot be set if it is cleared in this register. See Section 3.3
0x17	INT_MASK2	RW	11	yes	Interrupt mask bit field for INT_EVENT2. A bit in INT_EVENT2 cannot be set if it is cleared in this register. See Section 3.3
0x18	INT_CLEAR1	RW	11	yes	Interrupt clear bit field for INT_EVENT1. Bits set in this register are cleared from INT_EVENT1. See Section 3.3
0x19	INT_CLEAR2	RW	11	yes	Interrupt clear bit field for INT_EVENT2. Bits set in this register are cleared from INT_EVENT2.
0x1A	STATUS	RO	5	yes	Status bit field for non-interrupt events. See Section 3.4
0x1B-0x1E	Reserved				
0x1F	Sx_CONFIG	RO	24	yes	Power state configuration. The Host may write the current system power state, and a change in power state triggers a new Application Configuration to be applied. Virtual App Config GPIO are also triggered in this register. See Section 3.5
0x20	SET_Sx_APP_CONFIG	RO	2	yes	Configuration based on system state. The Host may write the current system power state, and a change in power state triggers a new Application Configuration to be applied. Virtual App Config GPIO are also triggered in this register. See Section 3.6
0x21	DISCOVERED_SVIDS	RO	33	yes	Received Discover SVID ACK message(s). This register contains the SVID information returned from Discover SVIDs REQ messages. See Section 3.7
0x22	Reserved				

Table 1-1. Unique Address Interface Registers (continued)

Register Number(1)	Register Name	Access	# Data Bytes	Unique per Port	Description
0x23	USB_CONFIG	RW	4	yes	USB configuration. This register is used to formulate Enter_USB message when applicable. See Section 3.8
0x24	USB_STATUS	RO	9	yes	USB status. This register provides status of transmitted or received Enter_USB message when applicable. See Section 3.9
0x25	Reserved				
0x26	POWER_PATH_STATUS	RO	5	no	Power Path Status. See Section 3.10
0x27	GLOBAL_SYSTEM_CONFIG	RW	10	no	Global system configuration (all ports). This register contains configuration bits that define hardware that is common to all ports and in most cases will not change in normal operation or will not require immediate action if changed. Any modifications to this register will cause a port disconnect and reconnect with the new settings. Initialized by Application Customization. See Section 3.11
0x28	PORT_CONFIG	RW	3	yes	Configuration for port-specific hardware. This register configures hardware that is specific for each port and in most cases will not change in normal operation or will not require immediate action if changed. Any modifications to this register will cause a port disconnect and reconnect with the new settings. Initialized by Application Customization. See Section 3.12
0x29	PORT_CONTROL	RW	4	yes	Configuration bits affecting system policy. These bits may change during normal operation and are used for controlling the respective port. The PD Controller will not take immediate action upon writing. Changes made to this register will take effect the next time the appropriate policy is invoked. Initialized by Application Customization. See Section 3.13
0x2A	Reserved				
0x2B	Reserved				
0x2C	Reserved				
0x2D	BOOT_STATUS	RO	5	no	Detailed status of boot process. This register provides details on PD Controller boot flags, Customer OTP configuration, and silicon revision See Section 3.14
0x2E	BUILD_DESCRIPTION	RO	49	no	Build description. This is an ASCII string that uniquely identifies custom build information.
0x2F	DEVICE_INFO	RO	40	no	Device information. This is an ASCII string with hardware and firmware version information of the PD Controller.
0x30	RX_SOURCE_CAPS	RO	29	yes	Received Source Capabilties. This register stores latest Source Capabilities message received over BMC. See Section 3.15
0x31	RX_SINK_CAPS	RO	29	yes	Received Sink Capabilities. This register stores latest Sink Capabilities message received over BMC. See Section 3.16
0x32	TX_SOURCE_CAPS	RW	31	yes	Source Capabilities for sending. This register stores PDOs and settings for outgoing Source Capabilities PD messages. Initialized by Application Customization. See Section 3.17
0x33	TX_SINK_CAPS	RW	29	yes	Sink Capabilities for sending. This register stores PDOs for outgoing Sink Capabilities USB PD messages. Initialized by Application Customization. See Section 3.18

Table 1-1. Unique Address Interface Registers (continued)

Register Number[(1)]	Register Name	Access	# Data Bytes	Unique per Port	Description
0x34	ACTIVE_CONTRACT_PDO	RO	6	yes	Power data object for active contract. This register stores PDO data for the current explicit USB PD contract, or all zeroes if no contract. See Section 3.19
0x35	ACTIVE_CONTRACT_RDO	RO	4	yes	Power data object for the active contract. This register stores the RDO of the current explicit USB PD contract, or all zeroes if no contract. See Section 3.20
0x36	Reserved				
0x37	AUTO_NEGOTIATE_SINK	RW	16	yes	Configuration for sink power negotiations. This register defines the voltage range between which the system can function properly, allowing the PD Controller to negotiate its own contracts. Initialized by Application Customization. See Section 3.21
0x38-0x3E	Reserved				
0x3F	POWER_STATUS	RO	2	yes	Details about the power of the connection. This register reports status regarding the power of the connection. See Section 3.22
0x40	PD_STATUS	RO	4	yes	Status of PD and Type-C state-machine. This register contains details regarding the status of PD messages and the Type-C state machine. See Section 3.23
0x41	PD3_STATUS	RO	4	yes	Status bit field for PD3.0 messages and state machine. See Section 3.24
0x42	PD3_CONFIG	RW	3	no	PD3.0 configuration settings. See Section 3.25
0x43	DELAY_CONFIG	RW	28	no	Optional delay configurations. See Section 3.26
0x44-0x46	Reserved				
0x47	TX_IDENTITY	RW	25	yes	Data to use for Discover Identity ACK. This data is sent in the response to Discover Identity REQ message. Initialized by Application Customization. See Section 3.27
0x48	RX_IDENTITY_SOP	RO	25	yes	Received Discover Identity ACK (SOP). Latest Discover Identity response received over USB PD using SOP. See Section 3.28
0x49	RX_IDENTITY_SOPp	RO	25	yes	Received Discover Identity ACK (SOP' or SOP''). Latest Discover Identity response received over USB PD using SOP'. See Section 3.29
0x4A	USER_VID_CONFIG	RW	63	yes	User VID Configuration. Initialized by Application Customization. See Section 3.30
0x4B	MIPI_VID_CONFIG	RW	4	yes	See Section 3.31
0x4C-0x4D	Reserved				
0x4E	RX_ATTENTION_VDM	RO	9	yes	Received Attention message. Latest Structured VDM Attention Initiator message received over USB PD. NOTE: Only Structured VDM "Attention" messages get stored in this buffer. See RX_OTHER_VDM register 0x4F for all other inbound VDMs. See Section 3.32
0x4F	RX_OTHER_VDM	RO	29	yes	Received VDM (not Attention). This register contains the latest VDM message received over USB PD except for Structured VDM Attention Initiator messages and SOP*_Debug messages. See Section 3.33
0x50	DATA_CONTROL	RW	6	yes	Data provided by the Thunderbolt Controller. See Section 3.34

Table 1-1. Unique Address Interface Registers (continued)

Register Number[1]	Register Name	Access	# Data Bytes	Unique per Port	Description
0x51	DP_SID_CONFIG	RW	6	yes	DisplayPort Alternate Mode configuration. Initialized by Application Customization See Section 3.35
0x52	INTEL_VID_CONFIG	RW	7	yes	Intel VID Configuration. This register also contains TBT Alternate mode configurations. Initialized by Application Customization. See Section 3.36
0x53	Reserved				
0x54	Reserved				
0x55	Reserved				
0x57	USER_VID_STATUS	RO	2	yes	User VID Status. See Section 3.37
0x58	DP_SID_STATUS	RO	37	yes	DisplayPort Alternate Mode Status. See Section 3.38
0x59	INTEL_VID_STATUS	RO	11	yes	Intel VID Thunderbolt Alternate Mode Status. See Section 3.39
0x5A	Reserved				
0x5B	Reserved				
0x5C	GPIO_CONFIG	RO	49	no	Application-specific GPIO Configurations. See Section 3.40
0x5D	RETIMER_DEBUG_MODE	RW	4	yes	See Section 3.41
0x5E	Reserved				
0x5F	DATA_STATUS	RO	5	yes	See Section 3.42
0x60	RX_USER_SVID_ATTN_VDM	RO	9	yes	Received Attention message for User VID. This register contains the latest Structured VDM Attention Initiator message received for User VID. See Section 3.43
0x61	RX_USER_SVID_OTHER_VDM	RO	29	yes	Latest Unstructured VDM or a non-Attention Structured VDM received for User VID. See Section 3.44
0x62	BINARYDATA_INDICES	RO	8	no	See Section 3.45
0x63	MIPI_VID_STATUS	RO	1	yes	See Section 3.46
0x64	I2CMASTER_CONFIG	RO	11	yes	See Section 3.47
0x65-0x68	Reserved				
0x69	TYPEC_STATE	RO	4	yes	Contains current status of both CCn pins. See Section 3.48
0x6A	ADC_RESULTS	RO	9[2]	no	Provides access to measurements from the internal ADC. See Section 3.49
0x6B	Reserved				
0x6C	EVENT_CONFIGURATION	RO	60	yes	Configures special actions to take when a User SVID mode is entered or exited. See Section 3.50
0x6D-0x6E	Reserved				
0x6F	Reserved				
0x70	SLEEP_CONFIG	RW	1	no	Sleep configurations. See Section 3.51
0x71	Reserved				
0x72	GPIO_STATUS	RO	8	no	Captures status and settings of all GPIO pins. See Section 3.52
0x73	TX_MIDB_SOP	RW	22	no	Transmit Manufacturer Info Data Block SOP (MIDB). See Section 3.53
0x74	RX_ADO	RO	4	yes	Received Alert Message as defined by USB PD. See Section 3.54

[2] TPS65994AD has length of 9 bytes. TPS65993AD has length of 9 bytes. TPS65992xAD has length of 11 bytes.

Table 1-1. Unique Address Interface Registers (continued)

Register Number[1]	Register Name	Access	# Data Bytes	Unique per Port	Description
0x75	TX_ADO	RW	4	no	Alert message to transmit as defined by USB PD. The PD controller transmits this data as-is from this register when 'ALRT' is issued. See Section 3.55
0x76	Reserved				
0x77	TX_SCEDB	RW	14	no	Transmit Source Capabilities Extended Data Block (SCEDB). See Section 3.56
0x78	Reserved				
0x79	TX_SDB	RW	6	yes	Transmit Status Data Block (SDB). See Section 3.57
0x7A	Reserved				
0x7B	TX_BSDO	RW	16	no	Transmit Battery Status Data Objects (BSDO). See Section 3.58
0x7C	Reserved				
0x7D	TX_BCDB	RW	36	no	Transmit Battery Capability Data Block (BCDB). See Section 3.59
0x7E	TX_SKEDB	RW	11	no	Transmit Sink Capabilities Data Block (SKEDB) See Section 3.60

The PD Controller implements the Unique Address Interface Tasks defined in Table 1-1.

Table 1-2. Unique Address Interface Tasks

Command 4CC	Type	Command Summary	References
AMDs	Alternate Mode	Start discovery process	See Section 4.5.3
AMEn	Alternate Mode	PD send Enter Mode	See Section 4.5.1
AMEx	Alternate Mode	PD send Exit Mode	See Section 4.5.2
GCdm	Alternate Mode	Get custom discovered modes	See Section 4.5.4
Gaid	CPU Control	Return to normal operation.	See Section 4.2.1
GAID	CPU Control	Cold reset request	See Section 4.2.2
FLad	Patch Bundle Update	External EEPROM Start Address	See Section 4.7.6
FLrd	Patch Bundle Update	External EEPROM Read	See Section 4.7.5
FLvy	Patch Bundle Update	External EEPROM Verify	See Section 4.7.8
FLwd	Patch Bundle Update	External EEPROM Memory Write	See Section 4.7.7
PBMs	Patch Bundle Update	Start Patch Burst Download Sequence	See Section 4.7.1
PBMc	Patch Bundle Update	Patch Burst Download Complete	See Section 4.7.2
PBMe	Patch Bundle Update	Patch Burst Mode Exit	See Section 4.7.3
GO2P	Patch Bundle Update	Forces PD controller to return to 'PTCH' mode and wait for patch over I2C.	See Section 4.7.4
ALRT	PD Message	Send a USB PD Alert message.	See Section 4.4.11
GPPI	PD Message	Send a USB PD Get* message.	See Section 4.4.7
GSkC	PD Message	PD Get Sink Capabilities	See Section 4.4.5
GSrC	PD Message	PD Get Source Capabilities	See Section 4.4.6
MBRd	PD Message	Read from PD message buffer.	See Section 4.4.10
SSrC	PD Message	PD Send Source Capabilities	See Section 4.4.8
SWDF	PD Message	PD DR_Swap to DFP	See Section 4.4.3
SWSk	PD Message	PD PR_Swap to Sink	See Section 4.4.1
SWSr	PD Message	PD PR_Swap to Source	See Section 4.4.2
SWUF	PD Message	PD DR_Swap to UFP	See Section 4.4.4
Trig	System	Emulate a GPIO input event	See Section 4.8.5

Table 1-2. Unique Address Interface Tasks (continued)

Command 4CC	Type	Command Summary	References
VDMs	PD Message	PD send VDM	See Section 4.5.5
ABRT	System	Abort current task	See Section 4.8.1
ANeg	System	Re-evaluate the auto-negotiate sink register	See Section 4.8.2
DBfg	System	Clear Dead Battery Flag	See Section 4.8.3
DISC	Modal	Simulate port disconnect	See Section 4.3.1
I2Cr	System	Executes I2C read transaction on I2C3m.	See Section 4.8.6
I2Cw	System	Executes I2C write transaction on I2C3m.	See Section 4.8.7
MuxR	System	Repeats transactions on I2C3m under certain conditions.	See Section 4.8.4
SRDY	Power Switch	System ready to sink power	See Section 4.6.1
SRYR	Power Switch	SRDY reset	See Section 4.6.2
UCSI	UCSI	Execute a UCSI command.	See Section 4.9.1
DRST	System	Execute a Data Reset per USB specifications	See Section 4.4.9

PD Controller Policy Modes

2.1 Overview

The PD Controller implements modes for "SRC Policy" (handing out Source contracts), modes for "SNK Policy" (issuing Requests for Sink contracts), and modes for "AM Policy" (Alternate Mode negotiation).

2.2 Source Policy Mode

The PD Controller uses the *TX_SOURCE_CAPS* register (0x32) to know what PDO(s) to advertise. The PD Controller will automatically respond to Request messages as appropriate, and each port acts independently. The host may dynamically change the *TX_SOURCE_CAPS* register, then issue the 'SSrC' 4CC Task and the PD controller will advertise the new PDO(s).

2.3 Sink Policy Mode

The PD Controller will always prepare its own Request message based on the settings in the *AUTO_NEGOTIATE_SINK* register (0x37) and the *TX_SINK_CAPS* register (0x33). The PD Controller will send its prepared Request message as soon as it is ready. The host may change the *AUTO_NEGOTIATE_SINK* register and/or the *TX_SINK_CAPS* register, then issue the 'GSrC' or 'ANeg' 4CC Task and the PD controller will re-negotiate the PD contract based on the updated values.

2.4 Alternate Modes and Alternate Mode Policy

Power Delivery enables alternative modes of operation by providing the mechanisms to discover, enter, and exit Alternate Modes. The PD Specification defines mechanisms to discover, enter and exit Modes defined either by a standard or by a particular vendor. These Modes can be supported either by the Port Partner or by a cable connecting the two Port Partners.

The PD controller will automatically send the Discover Identity message to both the Cable Plug and the Port Partner in compliance wtih USB PD rules. The responses are available in the *RX Identity SOP* register (0x48) and the *RX_IDENTITY_SOPp* register (0x49). The PD controller will also automatically send a Discover SVIDs message and store the response in the *DISCOVERED_SVIDS* register (0x21). The host should wait until *INT_EVENTx.DiscoverModesComplete* is asserted to read the information that was gathered. The host may configure other automatic behavior (even automatic mode entry) using the *USER_VID_CONFIG* register (0x4A), *DP_SID_CONFIG* register (0x51), and *INTEL_VID_CONFIG* register (0x52).

When allowed by the USB PD specification, the PD controller will respond to SVDM commands as appropriate. The *TX_IDENTITY* register (0x47) is used to formulate the response when a Discover Identity SVDM is received. When an SVDM is received it is stored in the *RX_ATTENTION_VDM* register (0x4E) or *RX_OTHER_VDM* register (0x4F), and either *INT_EVENTx.AttentionReceived* or *INT_EVENTx.VDMReceived* will get assserted.

The host may use the 4CC Tasks listed in Section 4.5 to implement other VDM behaviors.

Chapter 3

SLVUBT3–May 2020

Unique Address Interface Register Detailed Descriptions

3.1 0x03 MODE Register

The Mode register is a 32-bit register that returns 4 ASCII characters.

Table 3-1. 0x03 MODE Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x03	MODE	RO	4	no	See table below.

Table 3-2. 0x03 MODE Register Bit Field Definitions

Bits	Name	Description
31:0	mode	The mode described in 4 ASCII characters. See table below for more details.

Table 3-3. Description of device modes.

<table>
<tr><th>Mode register value</th><th>Description</th><th>$\overline{\text{I2C\_EC\_IRQ}}$ may be asserted</th><th>$\overline{\text{I2C2s\_IRQ}}$ may be asserted</th><th>Register availability</th></tr>
<tr><td>'APP '</td><td>The PD Controller is fully functioning in the application firmware.</td><td>Yes</td><td>Yes</td><td>All registers fully available.</td></tr>
<tr><td>'BOOT'</td><td>Device booting in dead battery.</td><td>No</td><td>No</td><td>Limited (see list at the end of this table)</td></tr>
<tr><td>'PTCH'</td><td>Device in patch mode.</td><td>Yes</td><td>No</td><td>Limited (see list at the end of this table)</td></tr>
<tr><td>Any other value</td><td>The PD Controller is functioning in a limited capacity.</td><td>No</td><td>No</td><td>No</td></tr>
<tr><td colspan="5">The host should not read or write most registers while the device is in the 'BOOT' or 'PTCH' mode. Only the following registers are available in 'BOOT' and 'PTCH' modes:
• the 4CC patch commands
• the 'GAID' and 'Gaid' 4CC commands
• VID (0x00), DID (0x01), PROTOVER (0x02), and UID (0x05)
• MODE (0x03)
• TYPE (0x04), VERSION (0x0F)
• CMD1 (0x08), DATA1 (0x09)
• DEVICE_CAPABILITIES (0x0D)
• INT_EVENT1 (0x14), INT_MASK1 (0x16), and INT_CLEAR1 (0x18)
• BOOT_STATUS (0x2D)
• DEVICE_INFO (0x2F)
• DATA_STATUS (0x5F)</td></tr>
</table>

3.2 0x0D DEVICE_CAPABILITIES Register

Table 3-4. 0x0D DEVICE_CAPABILITIES Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x0D	DEVICE_CAPABILITIES	RO	4	no	

Table 3-5. 0x0D DEVICE_CAPABILITIES Register Bit Field Definitions

Bits	Name	Description	
31:8	Reserved		
7	I2C3mLevel	Pull-up voltage required for I2C3m port.	
		0b	1.8V or 3.3V
		1b	3.3V
6:5	BC1p2Supported	BC 1.2 support capability.	
		00b	Not supported.
		01b	Only source supported.
		10b	Reserved
		11b	Source and sink supported.
4	SinglePort	Number of USB ports supported.	
		0b	The device is dual-port capable.
		1b	The device only has one port.
3	TbtPresent	Thunderboth support capability.	
		0b	Not supported.
		1b	Supported.
2	UsbPdCapability	USB Power Delivery capability.	
		0b	Supported
		1b	Not supported.
1:0	PowerRole	Power Role capability.	
		00b	Both source and sink roles supported (DRP).
		01b	Source-only.
		10b	Sink-only.
		11b	Source-only.

3.3 0x14 - 0x19 INT_EVENTX, INT_MASKX, INT_CLEARX Registers

Bytes 1 to 10 of this register are port-specific, but Byte 11 is common to all ports in the PD controller.

Table 3-6. 0x14 - 0x19 INT_EVENTX, INT_MASKX, INT_CLEARX Registers

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x14	INT_EVENT1	RO	11	yes	0
0x15	INT_EVENT2	RO	11	yes	0
0x16	INT_MASK1	RW	11	yes	Initialized by Application Configuration
0x17	INT_MASK2	RW	11	yes	Initialized by Application Configuration
0x18	INT_CLEAR1	RW	11	yes	0
0x19	INT_CLEAR2	RW	11	yes	0

Table 3-7. 0x14 - 0x19 INT_EVENTX, INT_MASKX, INT_CLEARX Registers Bit Field Definitions

Bits	Name	Description
Byte 11: Patch Status (common to all slave ports)		
7:3	Reserved	
2	I2CMasterNACKed	A transaction on the I2C master was NACKed.
1	ReadyForPatch	Device ready for a patch bundle from the host.
0	PatchLoaded	Patch was loaded to the device.
Bytes 9-10:		
15	AlertMessageReceived	Alert Message received, see RX_ADO register.
14	ChunkRequestReceived	Chunk Request message received.
13	ChunkResponseReceived	Chunk Response message received.
12	FrsSignalReceived	FRS swap signaling received.
11:9	Reserved	
8	AMDRenoirMuxError	An error occured configuring the AMD Renoir Mux. The configuration is done using the I2C3m port.
7	NotSupportedReceived	A NOT_SUPPORTED USB PD message was received. The EC should clear this bit before using a 4CC task that could result in receiving a Not_Supported message.
6	EventSocAckTimeout	The SoC has waited too long to process an alert.
2	MBRdBufferReady	Receive memory buffer full and ready to be read using the 'MBRd' command.
1	TXMemBufferEmpty	Transmit memory buffer empty.
0	PD3StatusUpdate	Asserted when the contents of PD3_STATUS register (0x41) change.
Bytes 5-8:		
31	IntelVIDStatusUpdate	Asserted when the contents of INTEL_VID_STATUS register (0x59) change.
30	DPSIDStatusUpdate	Asserted when the contents of DP_SID_STATUS register (0x58) change.
29:28	Reserved	
27	UserSVIDOtherVdmReceived	A User VID SVDM non-Attention or unstructured VDM has been received.
26	UserSVIDAttnVdmReceived	A User VID SVDM Attention has been received.
25	UserSVIDModeExited	A User VID alternate mode has been exited.
24	UserSVIDModeEntered	A User VID alternate mode has been entered.
23:22	Reserved	
21	DataResetStart	Set when the Data Reset process has started. The system is expected to handle the data-line terminations properly.
20	ExitModecomplete	Set when the Exit Mode process is complete.

Table 3-7. 0x14 - 0x19 INT_EVENTX, INT_MASKX, INT_CLEARX Registers Bit Field Definitions (continued)

Bits	Name	Description
19	DiscoverModeComplete	Set when the Discover Modes process has completed.
18	Reserved	
17	AMEntered	Set when any alternate mode is entered.
16	AMEntryFail	Set when any alternate mode attempted and failed.
15	Reserved	
14	ErrorUnableToSource	The Source was unable to increase the voltage to the negotiated voltage of the contract.
13	Reserved	
12	ProcHotNotification	A prochot event has occurred.
11	PlugEarlyNotification	A connection has been detected but not debounced.
10	SnkTransitionComplete	This event only occurs when in source mode (PD_STATUS.PresentPDRole = 1b). It occurs tSrcTransition (ms) after sending an Accept message to a Request message, just before sending the PS_RDY message.
9	Reserved	
7	ErrorMessageData	An erroneous message was received.
6	ErrorProtocolError	An unexpected message was received from the partner device.
5	Reserved	
4	ErrorMissingGetCapMessage	The partner device did not respond to the Get_Sink_Cap or Get_Source_Cap message that was sent.
3	ErrorPowerEventOccurred	An OVP, or ILIM event occurred on VBUS. Or a TSD event occurred. ILIM does not cause PORT to enter ErrorRecovery if the power path is DCDC.
2	ErrorCanProvideVoltageOrCurrentLater	The USB PD Source can provide acceptable voltage and current, but not at the present time. A "wait" message was sent or received.
1	ErrorCannotProvideVoltageOrCurrent	The USB PD Source cannot provide an acceptable voltage and/or current. A Reject message was sent to the Sink or a Capability Mismatch was received from the Sink.
0	ErrorDeviceIncompatible	When set to 1, a USB PD device with an incompatible specification version was connected. Or the partner device is not USB PD capable.
Bytes 1-4:		
31	Cmd2Complete	Set whenever a non-zero value in CMD2 register is set to zero or !CMD.
30	Cmd1Complete	Set whenever a non-zero value in CMD1 register is set to zero or !CMD.
29	Reserved	
28	Reserved	
27	PDStatusUpdate	Set whenever contents of PD_STATUS register (0x40) change.
26	StatusUpdate	Set whenever contents of STATUS register (0x1A) change.
25	DataStatusUpdate	Set whenever contents of DATA_STATUS register (0x5F) change.
24	PowerStatusUpdate	Set whenever contents of POWER_STATUS register (0x3F) change.
23	PPswitchChanged	Set whenever contents of POWER_PATH_STATUS register (0x26) changes.
22	Reserved	
21	UsbHostPresentNoLonger	Set when STATUS.UsbHostPresent transitions to anything other than 11b.
20	UsbHostPresent	Set when STATUS.UsbHostPresent transitions to 11b.
19	Reserved	

Table 3-7. 0x14 - 0x19 INT_EVENTX, INT_MASKX, INT_CLEARX Registers Bit Field Definitions (continued)

Bits	Name	Description
18	DRSwapRequested	A DR swap was requested by the Port Partner.
17	PRSwapRequested	A PR swap was requested by the Port Partner.
16	Reserved	
15	Reserved	
14	SourceCapMsgRcvd	This is asserted when a Source Capabilities message is received from the Port Partner.
13	NewContractAsProv	An RDO from the far-end device has been accepted and the PD Controller is a Source. This is asserted after the PS_RDY message has been sent. See ACTIVE_CONTRACT_PDO register (0x34) and ACTIVE_CONTRACT_RDO register (0x35) for details.
12	NewContractAsCons	Far-end source has accepted an RDO sent by the PD Controller as a Sink. See ACTIVE_CONTRACT_PDO register (0x34) and ACTIVE_CONTRACT_RDO register (0x35) for details.
11	VDMReceived	A Vendor Defined Message has been received. See RX_OTHER_VDM register (0x4F) for details.
10	AttentionReceived	An Attention Message has been received. See RX_ATTENTION_VDM register (0x4E) for details.
9	Overcurrent	Set whenever an Overcurrent field (VBUS or VCONN) in the POWER_PATH_STATUS register (0x26) changes.
8	Reserved	
7	SourceCapUpdated	The Source Capabiliities has been updated based on some automatic behavior configured by or requested by the host. The next Source Capabilities PD message transmitted will contain the updated values. This could get asserted when GLOBAL_SYSTEM_CONFIG.EnableSPM is asserted or when the 4CC Task 'UCSI' is used to implement SET_POWER_LEVEL. UCSI is not applicable in Blackjack/Pontoon
6	FRSwapComplete	A Fast Role swap has completed.
5	DRSwapComplete	A Data Role swap has completed. See STATUS register (0x1A) and PD_STATUS register (0x40) for port state.
4	PRSwapComplete	A Power role swap has completed. See STATUS register (0x1A) and PD_STATUS register (0x40) for port state.
3	PlugInsertOrRemoval	USB Plug Status has Changed. See Status register for more plug details.
2	Reserved	
1	PDHardReset	A PD Hard Reset has been performed. See PD_STATUS.HardResetDetails for more information.
0	Reserved	

3.4 0x1A STATUS Register

Table 3-8. 0x1A STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x1A	STATUS	RO	5	yes	Never fully reset though many bits change during disconnect and connect.

Table 3-9. 0x1A STATUS Register Bit Field Definitions

Bits	Name	Description	
Byte 5:			
7:2	Reserved		
1:0	AMStatus	Status of alternate mode negotiations.	
		00b	No Alternate Modes attempted.
		01b	At least one Alternate Mode entry successful. In addition, no Alternate Mode entries were unsuccessful.
		10b	At least one Alternate Mode entry unsuccessful. In addition no Alterate Mode entries were successful.
		11b	Some Alt Mode entries successful and some failed. At least one Alternate Mode entry successful and at least one mode entry unsuccessful.
Bytes 1-4:			
31	Reserved		
30	SocAckTimeout	Indicates whether the attached SoC has responded timely.	
		0b	SoC has responded timely.
		1b	SoC has not responded timely. SoC needs to respond to assertion of INT_EVENTx.DataStatusUpdate.
29	Reserved		
28	Reserved		
27	Bist	Indicates if a BIST procedure is in progress.	
		0b	No BIST in progress.
		1b	BIST in progress. This may also be indicated by MODE register (0x03) reading 'BIST'.
26	Reserved		
25:24	ActingAsLegacy	Indicates when PD Controller has gone into a mode where it is acting like a legacy (non PD) device. It can take approximately 10 seconds for the PD controller to determine that it is attached to a legacy source or sink.	
		00b	PD Controller is not in a legacy (non PD) mode
		01b	PD Controller is acting like a legacy sink. It will not respond to USB PD message traffic.
		10b	PD Controller is acting like a legacy source. It will not respond to USB PD message traffic.
		11b	PD controller is acting as a legacy sink (non-PD) port until the dead battery flag is cleared. The PD controller enters this state if no Source Capabilities are received after the boot process is complete. Once the dead-battery flag is cleared, the PD controller will send a Hard Reset.
23:22	UsbHostPresent	USB host attachment status.	
		00b	No host present. This means that no far-end device is presently providing VBUS or the PD Controller power role is Source.
		01b	VBUS is being provided by a Port Partner that is a PD device not capable of USB communications.
		10b	VBUS is being provided by a Port Partner that is not a PD device.
		11b	Host present. This means VBUS is being provided by a Port Partner that is USB PD capable and also capable of USB communications.

Table 3-9. 0x1A STATUS Register Bit Field Definitions (continued)

Bits	Name	Description	
21:20	VbusStatus	Indicates the present state of VBUS.	
		00b	At vSafe0V (less than 0.8V)
		01b	At vSafe5V (4.75V to 5.5V).
		10b	Within expected limits. The limits are determined based on the USB PD negotiated value.
		11b	Not within any of the other specified ranges.
19:7	Reserved		
6	DataRole	PD controller data role. This is only valid once there is a connection.	
		0b	Upward-facing port (UFP)
		1b	Downward-facing port (DFP)
5	PortRole	Current state of PD Controller CCx terminations. This also indicates the PD Controller Power Role, once connected. This bit does not toggle during Unattached.* state transitions.	
		0b	PD Controller is in the Sink role. This means the CCx pull-down is active or the port is disabled/disconnected.
		1b	PD Controller is Source (CCx pull-up active).
4	PlugOrientation	Plug orientation indicator. Indicates port orientation when known (requires connection).	
		0b	Upside-up orientation (plug CC on CC1). Could also be an unknown orientation or the port may be disabled/disconnected.
		1b	Upside-down orientation (plug CC on CC2).
3:1	ConnState	Details of a connected plug.	
		000b	No connection
		001b	Port is disabled
		010b	Audio connection (Ra/Ra)
		011b	Debug connection (Rd/Rd)
		100b	No connection, Ra detected (Ra but no Rd)
		101b	Reserved (may be used for Rp/Rp Debug connection)
		110b	Connection present, no Ra detected. Could be an Rd (but no Ra) or an Rp detected with no previous Ra detection, includes PD Controller that connected in Attached.SNK.
		111b	Connection present, Ra detected. Could be Rd (and Ra) detected or Rp detected (with previous Ra detection, if the PD Controller started as Source and later swapped to Sink).
0	PlugPresent	Status of the plug	
		0b	No plug is connected.
		1b	A plug is connected.

3.5 0x1F Sx_CONFIG Register

Table 3-10. 0x1F Sx_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x1F	Sx_CONFIG	RO	24	yes	Never fully reset though many bit change during disconnect and connect.

Table 3-11. 0x1F Sx_CONFIG Register Bit Field Definitions

Bits	Name	Description
Bytes 19-24: Configuration for S5		
47:4	Reserved	Reserved
3:1	S5_ConfigAddress	Port1 virtual address of S5 config. This config will be loaded in S5.
0	S5_ConfigEnable	Enables App Config loading upon entry to S5.
Bytes 13-18: Configuration for S4		
47:4	Reserved	Reserved
3:1	S4_ConfigAddress	Port1 virtual address of S4 config. This config will be loaded in S4.
0	S4_ConfigEnable	Enables App Config loading upon entry to S4.
Bytes 7-12: Configuration for S3		
47:4	Reserved	Reserved
3:1	S3_ConfigAddress	Port1 virtual address of S3 config. This config will be loaded in S3.
0	S3_ConfigEnable	Enables App Config loading upon entry to S3.
Bytes 1-6: Configuration for S0		
47:4	Reserved	Reserved
3:1	S0_ConfigAddress	Port1 virtual address of S0 config. This config will be loaded in S0.
0	S0_ConfigEnable	Enables App Config loading upon entry to S0.

3.6 0x20 SET_Sx_APP_CONFIG Register

Table 3-12. 0x20 SET_Sx_APP_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x20	SET_Sx_APP_CONFIG	RO	2	yes	Never fully reset though many bit change during disconnect and connect.

Table 3-13. 0x20 SET_Sx_APP_CONFIG Register Bit Field Definitions

Bits	Name	Description	
Byte 2: Reserved			
Byte 1: Sleep State			
7:3	Reserved	Reserved	
2:0	SleepState	Current sleep state. When a change in sleep state occurs, a new app config will be applied per the settings in 0x1F	
		000b	S0
		001b	S3
		010b	S4
		011b	S5
		100b-111b	Reserved

3.7 0x21 DISCOVERED_SVIDS Register

Table 3-14. 0x21 DISCOVERED_SVIDS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x21	DISCOVERED_SVIDS	RO	33	yes	Never fully reset though many bit change during disconnect and connect.

Table 3-15. 0x21 DISCOVERED_SVIDS Register Bit Field Definitions

Bits	Name	Description
Bytes 18-33: SOP' Response		
127:112	SvidsSopPrime7	Eighth SVID supported by SOP' cable plug
111:96	SvidsSopPrime6	Seventh SVID supported by SOP' cable plug
95:80	SvidsSopPrime5	Sixth SVID supported by SOP' cable plug
79:64	SvidsSopPrime4	Fifth SVID supported by SOP' cable plug
63:48	SvidsSopPrime3	Fourth SVID supported by SOP' cable plug
47:32	SvidsSopPrime2	Third SVID supported by SOP' cable plug
31:16	SvidsSopPrime1	Second SVID supported by SOP' cable plug
15:0	SvidsSopPrime0	First SVID supported by SOP' cable plug
Bytes 2-17: SOP Response		
127:112	SvidsSop7	Eighth SVID supported by SOP port partner
111:96	SvidsSop6	Seventh SVID supported by SOP port partner
95:80	SvidsSop5	Sixth SVID supported by SOP port partner
79:64	SvidsSop4	Fifth SVID supported by SOP port partner
63:48	SvidsSop3	Fourth SVID supported by SOP port partner
47:32	SvidsSop2	Third SVID supported by SOP port partner
31:16	SvidsSop1	Second SVID supported by SOP port partner
15:0	SvidsSop0	First SVID supported by SOP port partner
Byte 1: Header		
7:4	NumberSvidsSopPrime	Number of SVIDs discovered on SOP'
3:0	NumberSvidsSop	Number of SVIDs discovered on SOP.

3.8 0x23 USB_CONFIG Register

If the host requires changes to this register to take effect for an existing connection it must issue the 'DRST' 4CC task after writing the USB_CONFIG register. The recommended sequence is as follows:

- Write the changes to the USB_CONFIG register.
- Issue the 'DRST' 4CC Task.
- The PD controller will follow the USB data reset process.
- Once the data reset is complete, the PD controller will use the latest values in the USB_CONFIG register to formulate any Enter_USB message it sends.

Table 3-16. 0x23 USB_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x23	USB_CONFIG	RW	4	yes	Initialized by Application Configuration

Table 3-17. 0x23 USB_CONFIG Register Bit Field Definitions

Bits	Name	Description
31:27	Reserved	
26	Usb4Drd	Dual-role data capability for USB4. Assert if capable of operating as a USB4 device. Note that the USB PD spec requirements dictate that if this bit is asserted, the system also has to be device capable for USB 2.0.
25	Usb3Drd	Dual-role data capability for USB3. Assert if capable of operating as a USB3 device.
24:17	Reserved	
16	PCIe_Supported	PCIe control. This is used to set the "PCIe Support" bit when transmitting the Enter_USB message.
15	DP_Supported	DP support control. This is used to set the "DP Support" bit when transmitting the Enter_USB message.
14	TBT3_Supported	TBT3 support control. This is used to set the "TBT Support" bit when transmitting the Enter_USB message.
13	HostPresent	Host present control. This is used to set the "Host Present" bit when transmitting the Enter_USB message.
12:0	Reserved	

3.9 0x24 USB_STATUS Register

Table 3-18. 0x24 USB_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x24	USB_STATUS	RO	9	yes	

Table 3-19. 0x24 USB_STATUS Register Bit Field Definitions

Bits	Name	Description	
Bytes 6-9: vPro EnterMode VDO (transmitted or received)			
31:0	TbtEnterModeVdoDataRxTx	vPro mode VDO. This is the VDO received by the UFP or transmitted by the DFP when vPro mode was entered.	
Bytes 2-5: EUDO from Enter_USB message (transmitted or received)			
31:0	Usb4EnterUsbRxTx	Enter_USB Data Object (EUDO). This follows the USB PD definition for EUDO, it is sent by the DFP and received by the UFP.	
Byte 1: USB4 Mode Information			
7	Reserved		
6	USBRentryNeeded	Asserted when USB Re-entry is needed.	
5	vProEntryfailed	vPro mode error. This bit is asserted if an error occurred while trying to enter the vPro mode.	
4	Usb4ModeActiveOnPlug	USB4 mode status with Cable Plug. This bit is asserted when the mode required for USB4 has been entered on Cable Plug.	
3:2	Usb4RequiredPlugMode	USB4 mode requirement for Cable Plug. This field indicates which mode must be entered on the Cable Plug to enable USB4 mode.	
		00b	None
		01b	Reserved
		10b	USB4
		11b	TBT3
1:0	EUDOSOPSentOrReceived	USB4 status indicator with Port Partner. This bit is asserted while the USB4 mode is active with Port Partner.	
		00b	No Enter_USB. Enter_USB has not been sent as DFP, and has not been received as UFP.
		01b	Enter_USB timeout. The UFP should have received an Enter_USB, but after tUSB4Timeout it still has not been received.
		10b	Enter_USB failure. In DFP mode, the transmitted Enter_USB message received no response, a Reject message or a Not_Supported message. In UFP mode, a Reject message was sent in response to a received Enter_USB message.
		11b	Successful Enter_USB. An Enter_USB was transmitted or received as expected.

3.10 0x26 POWER_PATH_STATUS Register

Table 3-20. 0x26 POWER_PATH_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x26	POWER_PATH_STATUS	RO	5	no	0

Table 3-21. 0x26 POWER_PATH_STATUS Register Bit Field Definitions

Bits	Name	Description	
Bytes 4-5: PP and PP_CABLE over-current			
15:14	PowerSource	Indicates current PD Controller power source. NOTE: Since the Dead Battery flag forces PD Controller to be powered from VBUS, only 10b is valid when this flag is set. Any other setting indicates that the Dead Battery flag is not set.	
		00b	Reserved
		01b	PD Controller is powered from VIN_3V3.
		10b	PD Controller is powered from VBUS. The Dead Battery flag is set.
		11b	Reserved
13:12	Reserved	Reserved.	
11	PP_CABLE2_Overcurrent[1][2]	PP_CABLE2 overcurrent indicator. Asserted if an overcurrent condition exists on PP_CABLE2 (VCONN).	
10	PP_CABLE1_Overcurrent	PP_CABLE1 overcurrent indicator. Asserted if an overcurrent condition exists on PP_CABLE1 (VCONN).	
9:6	Reserved	Reserved.	
5	PP2_Overcurrent[1][2]	PP_5V2 overcurrent indicator. Asserted if an overcurrent conditions exists on PP2 switch (PP_5V2).	
4	PP1_Overcurrent	PP_5V1 overcurrent indicator. Asserted if an overcurrent conditions exists on PP1 switch (PP_5V1).	
3:0	Reserved		
Bytes 1-3: PP and PP_CABLE Switch Status			
17:15	PP4switch[1][2]	Indicates current state of PP4 (PP_EXT2).	
		0h	PP4 switch disabled.
		1h	The PP4 switch is currently disabled. It was disabled due to fault (system output).
		2h	Reserved
		3h	PP4 switch enabled (system input).
		4h-7h	Reserved
14:12	PP3switch	Indicates current state of PP3 (PP_EXT1).	
		0h	PP3 switch disabled.
		1h	PP3 switch currently disabled due to fault. The switch is a system output.
		2h	Reserved
		3h	PP3 switch enabled (system input).
		4h-7h	Reserved
11:9	PP2switch[1][2]	Indicates current state of PP2 switch (PP_5V2).	
		0h	PP2 switch disabled.
		1h	PP2 switch currently disabled due to fault. The switch is a system output.
		2h	PP2 switch enabled (system output).
		3h-7h	Reserved

[1] This feature is not supported by TPS65993AD_F509.04.02.

[2] This feature is not supported by TPS65992xAD_F509.05.02.

Table 3-21. 0x26 POWER_PATH_STATUS Register Bit Field Definitions (continued)

Bits	Name	Description	
8:6	PP1switch	Indicates current state of PP1 switch (PP_5V1).	
		0h	PP1 switch disabled.
		1h	PP1 switch currently disabled due to fault. The switch is a system output.
		2h	PP1 switch enabled (system output).
		3h-7h	Reserved
5:4	Reserved		
3:2	PP_CABLE2_switch[1][2]	Indicates current state of PP_CABLE2 switch.	
		00b	PP_CABLE2 switch disabled.
		01b	PP_CABLE2 switch currently disabled. The PD controller is waiting for PP5V pin to go high.
		10b	PP_CABLE2 switch CC1 enabled (system output).
		11b	PP_CABLE2 switch CC2 enabled (system output).
1:0	PP_CABLE1_switch	Indicates current state of PP_CABLE1 switch.	
		00b	PP_CABLE1 switch disabled.
		01b	PP_CABLE1 switch currently disabled. The PD controller is waiting for PP5V pin to go high.
		10b	PP_CABLE1 switch CC1 enabled (system output).
		11b	PP_CABLE1 switch CC2 enabled (system output).

3.11 0x27 GLOBAL_SYSTEM_CONFIG Register

The register fields defined in this register are for settings that are common to both ports of the device. In most usage cases, these settings will not change in normal operation or will not require immediate action if changed.

NOTE: Any modifications to this register will cause all ports in the PD controller to disconnect and reconnect with the new settings.

Table 3-22. 0x27 GLOBAL_SYSTEM_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x27	GLOBAL_SYSTEM_CONFIG	RW	10	no	Initialized by Application Configuration

Table 3-23. 0x27 GLOBAL_SYSTEM_CONFIG Register Bit Field Definitions

Bits	Name	Description	
Bytes 9-10: I2C2s Slave Address Configuration			
15:8	Port2I2C2sSlaveAddress	Slave address for Port 2 on I2C2s.	
7:0	Port1I2C2sSlaveAddress	Slave address for Port 1 on I2C2s.	
Byte 8: Reserved			
Byte 7: Reserved			
Bytes 4-6: Miscellaneous configuration			
23:19	Reserved		
18	I2C3mVoltageThreshold	Pull-up voltage for I2C3m. This is set to 3.3V by default.	
		0b	1.8 V
		1b	3.3 V
17	I2C2sVoltageThreshold	Pull-up voltage for I2C2s. This is set to 1.8V by default.	
		0b	1.8 V
		1b	3.3 V
16	I2C_EcVoltageThreshold	Pull-up voltage for I2C_EC. This is set to 1.8V by default.	
		0b	1.8 V
		1b	3.3 V
15	MinimumCurrentAdvertisement	Configuration for SPM. If the PD controller is configured to automatically reduce the current advertisement, it reduces to this value.	
		0b	0.9 A (USB default during an implicit contract)
		1b	1.5 A
14	EmulateSinglePort[1][2]	Assert this bit to disable one port. If this bit is asserted the PD controller will only support one Type-C port (Port A).	
13	DisableEepromUpdates	EEPROM updates not allowed if this bit asserted.	

[1] This feature is not supported by TPS65993AD_F509.04.02.

[2] This feature is not supported by TPS65992xAD_F509.05.02.

Table 3-23. 0x27 GLOBAL_SYSTEM_CONFIG Register Bit Field Definitions (continued)

Bits	Name	Description	
12:10	I2CTimeout	I2C bus timeout. The PD controller will reset its I2C slave hardware if for some reason it holds the SCL line low for more than the time selected in this register. This prevents locking up the I2C bus. This applies to both I2C1s_SCL and I2C2s_SCL.	
		0h	25 ms
		1h	50 ms
		2h	75 ms
		3h	100 ms
		4h	125 ms
		5h	150 ms
		6h	175 ms
		7h	1000 ms
9	Reserved		
8:7	MultiPortSinkNonOverlapTime	Delay configuration for MultiPortSinkPolicy. Controls the amount of time a new Sink input path closes after the old Sink input path opens. This forms a break-before-make condition when controlling Sink paths from both ports. This is only applicable when MultiPortSinkPolicy = 01b and applies to externally controlled switch paths (PP3 or PP4). This feature may be deprecated in the future.	
		00b	1 ms
		01b	5 ms
		10b	10 ms
		11b	15 ms
6	EnableSPM	Enable bit for simple source power management. See Section 5.5. The SPM feature also requires that the Prevent_High_Current_Contract_Event GPIO Event be assigned to one of the GPIO's or that the host issue the 'Trig' 4CC task to initialize the Rp advertisement. Otherwise, the initial advertisement as configured in the MinimumCurrentAdvertisement field will persist indefinitely.	
5	EnableOneUFPPolicy	Enable bit for simple UFP policy manager.	
4:2	TBTcontrollerType	Type of TBT controller. Controls specific behavior for different TBT controllers. See documentation for each TBT controller for details.	
		0h	Default
		1h	AR
		2h	TR or later Ridge platform. Platform contains a discrete TBT controller.
		3h	ICL or later Lake platform. Platform contains integrated TBT controller.
		4h-7h	Reserved
1	Reserved		
0	MultiPortSinkPolicy	Automatic sink-path coordination. This configures the how the PD controller controls input switches of each port when both ports are operating as a Sink.	
		0b	No Sink Management. Each Sink path behaves independently and do not consider the states of each other.
		1b	Highest Power. Only one Sink path is enabled at a time. If both ports are connected to a Source, the port with the higher power contract enables its Sink path.
Byte 3: Power Path Configuration 2			
7:6	RcpThreshold	Threshold used for RCP on PP_EXT.	
		00b	6 mV (nominal)
		01b	8 mV (nominal)
		10b	10 mV (nominal)
		11b	12 mV (nominal)

Table 3-23. 0x27 GLOBAL_SYSTEM_CONFIG Register Bit Field Definitions (continued)

Bits	Name	Description	
5:3	PP4Config[1][2]	PP4 configuration. This register configures PP4 switch controls.	
		0h	PP4 not used and disabled.
		1h	PP4 is a Source (output)
		2h	PP4 is a Sink (input)
		3h	PP4 is sink, but waits for 'SRDY'. This allows the host to determine when PP3 is enable and disabled.
		4h	PP3 is bi-directional. It may act as Sink (input) or Source (output).
		5h	PP4 is bi-directional, but waits for 'SRDY'. This allows the host to determine when PP3 is enable and disabled.
		6h-7h	Reserved
2:0	PP3Config	PP3 configuration. This register configures PP3 switch controls.	
		0h	PP3 not used and disabled.
		1h	PP3 is a Source (output)
		2h	PP3 is a Sink (input)
		3h	PP3 is sink, but waits for 'SRDY'. This allows the host to determine when PP3 is enable and disabled.
		4h	PP3 is bi-directional. It may act as Sink (input) or Source (output).
		5h	PP3 is bi-directional, but waits for 'SRDY'. This allows the host to determine when PP3 is enable and disabled.
		6h-7h	Reserved
Byte 2: Internal Power Path Configuration			
7:6	IlimOvershoot	PP_5V ILIM configuration. Controls the amount of overshoot used by the FW to select the current limit for the PP5V to Px_VBUS.	
		00b	No additional overshoot margin.
		01b	Overshoot margin of at least 100 mA.
		10b	Overshoot margin of at least 200 mA.
		11b	Reserved
5:3	PP2Config[1][2]	PP2 configuration (PP_5V2).	
		0h	Not used (disabled)
		1h	PP2 configured as source.
		2h-7h	Reserved
2:0	PP1Config	PP1 configuration (PP_5V1).	
		0h	Not used (disabled)
		1h	PP1 configured as source.
		2h-7h	Reserved
Byte 1: VCONN Power Path Configuration			
7:3	Reserved		
2	PPCable2Config[1][2]	Enable PP_CABLE2. If this bit is asserted the PD controller will enable VCONN on PP_CABLE2 when required for USB specification compliance.	
1	Reserved		
0	PPCable1Config	Enable PP_CABLE1. If this bit is asserted the PD controller will enable VCONN on PP_CABLE1 when required for USB specification compliance.	

3.12 0x28 PORT_CONFIG Register

The register fields defined in the PORT_CONFIG register are for settings of the hardware specific to the Type-C port used and, in most cases, will not change in normal operation or will not require immediate action if changed.

NOTE: Any modifications to this register will cause the addressed port in the PD controller to disconnect and reconnect with the new settings.

Table 3-24. 0x28 PORT_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x28	PORT_CONFIG	RW	3	yes	Initialized by Application Configuration

Table 3-25. 0x28 PORT_CONFIG Register Bit Field Definitions

Bits	Name	Description	
Byte 3: OVP and Soft Start Configuration			
7:6	Reserved	Reserved.	
5:4	OVP4PP5V	VBUS OVP settings while sourcing from PP5V. This applies while sourcing through PP1 or PP2. See data-sheet for voltage range.	
		00b	Use setting 0: 5.25 V (typical)
		01b	Use setting 1: 5.5 V (typical)
		10b	Use setting 2: 5.8 V (typical)
		11b	Use setting 3: 6.1 V (typical)
3:2	SoftStart	Soft start configuration settings. Controls the soft start for the sinking power path switch.	
		00b	0.41 V/ms (typical)
		01b	0.79 V/ms (typical)
		10b	1.57 V/ms (typical)
		11b	3.39 V/ms (typical)
1:0	VBUSOvpUsage	OVP configuration settings. These two bits are used to select the OVP trip-point. The PD controller automatically computes the lowest threshold that does not overlap with the expected maximum voltage (including maximum tolerance allowed by USB PD specification). The OVP trip-point will be set at the selected percentage of the computed threshold.	
		00b	100%.
		01b	105%.
		10b	111%.
		11b	114%.
Bytes 1-2: Port Configuration			
15	AMDI2CMuxEnable	AMD Renoir Mux I2C Master Enable.	
		0b	AMD I2C Mux is disabled on the I2C3 Master.
		1b	AMD I2C Mux is enabled on the I2C3 Master.
14:13	USB3rate	USB3 configuration.	
		00b	USB3 not supported.
		01b	USB3 Gen1 signaling rate supported.
		10b	USB3 Gen2 signaling rate supported.
		11b	Reserved.
12	Reserved		
11	UsbCommCapable	USB communications capable. Assert this bit in systems that are USB communications capable.	
10	DisablePD	Assert this bit to disable USB PD.	

Table 3-25. 0x28 PORT_CONFIG Register Bit Field Definitions (continued)

Bits	Name	Description	
9:8	TypeCSupportOptions	Configuration for optional features. This register controls whether optional Type-C state machine states are supported. NOTE: These states are mutually-exclusive and these options only exist when specific Type-C state machines are used.	
		00b	No Type-C optional states are supported.
		01b	Try.SRC state is supported as a DRP.
		11b	Reserved.
		Reserved. 7:2	Reserved
1:0	TypeCStateMachine	Port Configuration. This field will be overriden by SPM engine when SPM is enabled. It is not recommended to override SPM value. GUI must ensure HI to this register while SPM is running is not accesible. If one desires to override the SPM driven value, it is allowed however care must be taken to place the port in correct TypeC State for the desired operation. When this field is overriden by external controller, SPM does not take any action until the next SPM event.	
		00b	Sink state machine only.
		01b	Source state machine only.
		10b	DRP state machine.
		11b	Disabled. Type-C state-machine is disabled (CC pins are high-z).

3.13 0x29 PORT_CONTROL Register

Table 3-26. 0x29 PORT_CONTROL Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x29	PORT_CONTROL	RW	4	yes	Initialized by Application Configuration

Table 3-27. 0x29 PORT_CONTROL Register Bit Field Definitions

Bits	Name	Description	
31:30	ChargerDetectEnable[1][2]	Configure the types of legacy chargers to detect.	
		00b	Do not detect any legacy chargers.
		01b	Detect BC 1.2 chargers.
		10b	Reserved, do not use
		11b	Detect BC 1.2 and proprietary legacy chargers.
29	UsbDisable	Overrides USB connections in DATA_STATUS. If this bit is asserted, it forces USB2Connection and USB3Connection in the DATA_STATUS (0x5F) register to be zero.	
28:26	ChargerAdvertiseEnable[1][2]	Configure the types of legacy chargers to emulate.	
		0h	Do not emulate any legacy charger. This means only the SDP mode will be used with legacy devices.
		1h	BC 1.2 CDP only.
		2h	BC 1.2 DCP only.
		3h	Reserved
		4h	Reserved
		5h	DCP Auto 1 (2.7V and DCP)
		6h	DCP Auto 2 (1.2V, 2.7V and DCP)
		7h	Reserved
24	Resistor15kPresent[1][2]	Configure D+ and D- termination. Assert this bit if there is a 15kOhm pull-down on D+ and D- (USB2.0 Host Phy pull-downs enabled). This should not be used for DCP or DCP Auto modes.	
		0b	System does NOT have 15 kOhm pull-down. The PD controller will apply 15 kOhm pull-downs as necessary for BC 1.2.
		1b	System has 15 kOhm pull-down. The PD controller will not apply 15 kOhm pull-downs.
23	VconnCurrentLimit	Current limit configuration for PP_CABLEx.	
		0b	410 mA (typical)
		1b	590 mA (typical)
22	FRSwap_Enabled	Enable Fast-Role Swap as initial sink. Assert this bit to enable Fast-Role Swap detection. If the amount of current the Port Partner requires during Fast-Role Swap process is less than or equal to the current in TX_SOURCE_CAPS.TXSourcePDO1 when an FRS signal is detected, then the device will perform a fast-role swap.	
21	SinkControlBit	Configure reaction to UnconstrainedPower bit. This configures whether the state of the UnconstrainedPower bit affects PP3 or PP4.	
		0b	No affect on sink switches. The UnconstrainedPower (bit 19) does not affect sink switches. Barrel_Jack_Event GPIO status does not affect sink switches.
		1b	Disable sink switches automatically. When the Unconstrained Power (bit 19) bit is 1, then PP3 and PP4 are disabled. If the Barrel_Jack_Event GPIO is enabled and its status is high then PP3 and PP4 are disabled.
20	Reserved		

[1] This feature is not supported by TPS65994AD_F509.04.02.
[2] This feature is not supported by TPS65993AD_F509.04.02.

Table 3-27. 0x29 PORT_CONTROL Register Bit Field Definitions (continued)

Bits	Name	Description	
19	UnconstrainedPower	External power configuration. This also sets the Unconstrained Power bit defined by USB PD. When this bit is changed from 1 to 0 the PD controller will not attempt a power role swap with the Port Partner. If a power role swap is desired the host should issue a 'SWSr' 4CC command.	
		0b	No external power. There is no external power besides VBUS (if present) for the system.
		1b	External power present. The system is receiving external power from a source other than VBUS.
18	ForceUSB3Gen1	Forced Gen1 operation	
		0b	DATA_STATUS.USB3Speed register will always report full capabilities.
		1b	DATA_STATUS.USB3Speed register will report USB3 Gen1 only
17	AMIntrusiveMode	Assert to allow host to manage Alt mode process. It is recommended that this bit be set to 0b so that the PD controller can automatically handle Alternate Mode entry and exit.	
		0b	Do not operate in Alternate Mode Intrusive mode.
		1b	Disable automatic mode entry. The PD controller will not issue any Enter Mode Structured VDM Commands automatically. The PD Controller will still issue Discover SVIDs and Discover Modes commands as appropriate for any SVIDs it supports in common with the UFP, but this bit blocks any automatic entry into Alternate Modes. The AMEn and AMEx commands allow manual control over Alternate Modes when this bit is asserted.
16	AutomaticIDRequest	Configure identity discovery for SOP. If this bit is asserted, the PD Controller will automatically issue Discover Identity VDM for all SOP types when appropriate.	
15	InitiateSwapToDFP	Configure DR_Swap to DFP initiation. If this bit is asserted, the PD Controller automatically initiates and sends DR_Swap requests to the Port Partner when appropriate if presently operating as UFP.	
14	ProcessSwapToDFP	Configure response to DR_Swap to DFP. If this bit is asserted, the PD Controller will automatically accept a DR_Swap request to become a DFP. Otherwise, the PD Controller will reject such a request.	
13	InitiateSwapToUFP	Configure DR_Swap to UFP initiation. If this bit is asserted, the PD Controller automatically initiates and sends DR_Swap requests to the Port Partner when appropriate if presently operating as DFP.	
12	ProcessSwapToUFP	Configure response to DR_Swap to UFP. If this bit is asserted, the PD Controller will automatically accept a DR_Swap request to become a UFP. Otherwise, the PD Controller will reject such a request.	
11	RetimerFWUpdate	Enable Retimer FW update. Setting to 1 causes the PD controller to enter the retimer FW update state. Setting to 0 exits the update state.	
10:8	Reserved		
7	InitiateSwapToSource	Configure PR_Swap to source initiation. If this bit is asserted, the PD Controller automatically initiates and sends PR_Swap requests to the Port Partner when appropriate if presently operating as Sink (C/P).	
6	ProcessSwapToSource	Configure response to PR_Swap to source. If this bit is asserted, the PD Controller will automatically accept a PR_Swap request to become a Source. Otherwise, the PD Controller will reject such a request.	
5	InitiateSwapToSink	Configure PR_Swap to sink initiation. If this bit is asserted, the PD Controller automatically initiates and sends PR_Swap requests to the Port Partner when appropriate if presently operating as Source (P/C).	
4	ProcessSwapToSink	Configure response to PR_Swap to sink. If this bit is asserted, the PD Controller will automatically accept a PR_Swap request to become a Sink. Otherwise, the PD Controller will reject such a request.	
3:2	Reserved		

Table 3-27. 0x29 PORT_CONTROL Register Bit Field Definitions (continued)

Bits	Name	Description	
1:0	TypeCCurrent	Type-C Current advertisement. This setting is ignored if a Source role is not enabled and active. This setting is also ignored during an explicit USB PD contract, where the Rp value is used for collision avoidance as required by the USB PD specification. Note that when PP5V is low, the FW will only use the default Type-C current regardless of the value in this field.	
		00b	USB Default Current
		01b	1.5 A
		10b	3.0 A
		11b	Reserved

3.14 0x2D BOOT_STATUS Register

Table 3-28. 0x2D BOOT_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x2D	BOOT_STATUS	RO	5	no	Context dependent (never reset)

Table 3-29. 0x2D BOOT_STATUS Register Bit Field Definitions

Bits	Name	Description	
Byte 5: Revision ID			
7:0	REV_ID	Revision ID for the PD controller.	
Bytes 1-4: Boot Flags (treated as a 32-bit little endian value)			
31:29	PatchConfigSource	Source of patch configuration. This field indicates the source of the configuration patch that has been successfully loaded.	
		0h	No configuration has been loaded.
		1h	Source-only default configuration 00b.
		2h	Source-only default configuration 01b.
		3h	Source-only default configuration 10b.
		4h	Reserved.
		5h	A configuration has been loaded from EEPROM.
		6h	A configuration has been loaded from I2C.
		7h	Reserved.
28	Reserved		
27	Reserved		
24	Reserved		
19	MasterTSD	Master thermal shut-down indicator. This bit is asserted if the PD controller is rebooting after the master thermal sensor caused a reset.	
18	PP4switch	PP4 switch status. This bit is asserted when the PP4 sink path was enabled during dead-battery mode	
17	PP3switch	PP3 switch status. This bit is asserted when the PP3 sink path was enabled during dead-battery mode	
16:14	Reserved		
13	region1crcfail	Region 1 CRC status indicator. This bit is asserted when the CRC of data read from Region 1 of EEPROM memory failed.	
12	region0crcfail	Region 0 CRC status indicator. This bit is asserted when the CRC of data read from Region 0 of EEPROM memory failed.	
11	Reserved		
10	patchdownloaderr	Asserted when a patch download error occurs.	
9	region1eepromerr	Region 1 status indicator. This bit is asserted when an error occurred attempting to read Region 1 of EEPROM memory. A retry may have been successful.	
8	region0eepromerr	Region 0 status indicator. This bit is asserted when an error occurred attempting to read Region 0 of EEPROM memory. A retry may have been successful.	
7	region1invalid	Region 1 header status indicator. This bit is asserted when Region 1 header of the EEPROM memory was invalid.	
6	region0invalid	Region 0 header status indicator. This bit is asserted when Region 0 header of the EEPROM memory was invalid.	
5	region1	Region 1 attempted indicator. This bit is asserted when Region 1 of the EEPROM memory was attempted.	
4	region0	Region 0 attempted indicator. This bit is asserted when Region 0 of the EEPROM memory was attempted.	
3	I2cEepromPresent	EEPROM presence indicator. This bit is asserted when an EEPROM device was discovered on I2C3m during boot.	
2	DeadBatteryFlag	Dead Battery flag indicator. This bit is asserted when the PD Controller booted in dead-battery mode.	

Table 3-29. 0x2D BOOT_STATUS Register Bit Field Definitions (continued)

Bits	Name	Description
1	Reserved	Reserved.
0	PatchHeaderErr	Asserted when a patch bundle header errors.

3.15 0x30 RX_SOURCE_CAPS Register

Table 3-30. 0x30 RX_SOURCE_CAPS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x30	RX_SOURCE_CAPS	RO	29	yes	Cleared on disconnect, or Hard Reset.

Table 3-31. 0x30 RX_SOURCE_CAPS Register Bit Field Definitions

Bits	Name	Description
Bytes 26-29: PDO #7 (treated as a 32-bit little endian value)		
31:0	SourcePdo7	Seventh Source Capabilities PDO received
Bytes 22-25: PDO #6 (treated as a 32-bit little endian value)		
31:0	SourcePdo6	Sixth Source Capabilities PDO received
Bytes 18-21: PDO #5 (treated as a 32-bit little endian value)		
31:0	SourcePdo5	Fifth Source Capabilities PDO received
Bytes 14-17: PDO #4 (treated as a 32-bit little endian value)		
31:0	SourcePdo4	Fourth Source Capabilities PDO received
Bytes 10-13: PDO #3 (treated as a 32-bit little endian value)		
31:0	SourcePdo3	Third Source Capabilities PDO received
Bytes 6-9: PDO #2 (treated as a 32-bit little endian value)		
31:0	SourcePdo2	Second Source Capabilities PDO received
Bytes 2-5: PDO #1 (treated as a 32-bit little endian value)		
31:0	SourcePdo1	First Source Capabilities PDO received
Byte 1: Header		
7:3	Reserved	
2:0	numValidPDOs	Number of valid PDOs in this register. Each PDO is 4 bytes. (max of 7)

3.16 0x31 RX_SINK_CAPS Register

Table 3-32. 0x31 RX_SINK_CAPS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x31	RX_SINK_CAPS	RO	29	yes	Cleared on disconnect, or Hard Reset.

Table 3-33. 0x31 RX_SINK_CAPS Register Bit Field Definitions

Bits	Name	Description
Bytes 26-29: PDO #7 (treated as a 32-bit little endian value)		
31:0	SinkPdo7	Seventh Sink Capabilities PDO received
Bytes 22-25: PDO #6 (treated as a 32-bit little endian value)		
31:0	SinkPdo6	Sixth Sink Capabilities PDO received
Bytes 18-21: PDO #5 (treated as a 32-bit little endian value)		
31:0	SinkPdo5	Fifth Sink Capabilities PDO received
Bytes 14-17: PDO #4 (treated as a 32-bit little endian value)		
31:0	SinkPdo4	Fourth Sink Capabilities PDO received
Bytes 10-13: PDO #3 (treated as a 32-bit little endian value)		
31:0	SinkPdo3	Third Sink Capabilities PDO received
Bytes 6-9: PDO #2 (treated as a 32-bit little endian value)		
31:0	SinkPdo2	Second Sink Capabilities PDO received
Bytes 2-5: PDO #1 (treated as a 32-bit little endian value)		
31:0	SinkPdo1	First Sink Capabilities PDO received
Byte 1: Header		
7:3	Reserved	
2:0	numValidPDOs	Number of valid PDOs in this register. Each PDO is 4 bytes. (max of 7)

3.17 0x32 TX_SOURCE_CAPS Register

The PD controller will transmit Source Capabilities that are written to this register without verifying them (besides limiting current see below). The user is responsible to write this register correctly per the USB PD requirements. The PD controller will only use the first TXSourceNumPDOs PDO's, the host may write multiple PDO's during configuration then dynamically write TXSourceNumPDOs to change which PDO's are advertised. If this register is changed, the host must subsequently issue the 4CC command 'SSrC'. This will cause the PD controller to re-load this TX Source Capabilities register. If the host changes one of the PowerPathForPDOx fields, then the host must also issue the 4CC command 'DISC'. After reconnecting the PD controller will use the new values in the PowerPathForPDOx fields.

The PD controller will read the capabilities of the cable and limit the maximum current in each PDO to respect the cable's VBUS Current Handling Capability. See Table 3-128 for a description of how the maximum current allowed by the cable is determined.

Table 3-34. 0x32 TX_SOURCE_CAPS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x32	TX_SOURCE_CAPS	RW	31	yes	Initialized by Application Configuration

Table 3-35. 0x32 TX_SOURCE_CAPS Register Bit Field Definitions

Bits	Name	Description	
Bytes 28-31: PDO #7 (treated as a 32-bit little endian value)			
31:0	TXSourcePDO7	Seventh Source Capabilities PDO contents. See Table 3-37	
Bytes 24-27: PDO #6 (treated as a 32-bit little endian value)			
31:0	TXSourcePDO6	Sixth Source Capabilities PDO contents. See Table 3-37	
Bytes 20-23: PDO #5 (treated as a 32-bit little endian value)			
31:0	TXSourcePDO5	Fifth Source Capabilities PDO contents. See Table 3-37	
Bytes 16-19: PDO #4 (treated as a 32-bit little endian value)			
31:0	TXSourcePDO4	Fourth Source Capabilities PDO contents. See Table 3-37	
Bytes 12-15: PDO #3 (treated as a 32-bit little endian value)			
31:0	TXSourcePDO3	Third Source Capabilities PDO contents. See Table 3-37	
Bytes 8-11: PDO #2 (treated as a 32-bit little endian value)			
31:0	TXSourcePDO2	Second Source Capabilities PDO contents. See Table 3-37	
Bytes 4-7: PDO #1 (treated as a 32-bit little endian value)			
31:0	TXSourcePDO1	First Source Capabilities PDO contents. See Table 3-36	
Bytes 2-3: Power path configuration for each PDO.			
15:14	Reserved		
13:12	PowerPathForPDO7	Configures which PP to use for PDO7. Same format as PowerPathForPDO2.	
11:10	PowerPathForPDO6	Configures which PP to use for PDO6. Same format as PowerPathForPDO2.	
9:8	PowerPathForPDO5	Configures which PP to use for PDO5. Same format as PowerPathForPDO2.	
7:6	PowerPathForPDO4	Configures which PP to use for PDO4. Same format as PowerPathForPDO2.	
5:4	PowerPathForPDO3	Configures which PP to use for PDO3. Same format as PowerPathForPDO2.	
3:2	PowerPathForPDO2	Configures which PP to use for PDO2.	
		00b	Reserved
		01b	Reserved
		10b	PP_EXT1 is used for this PDO.
		11b	PP_EXT2 is used for this PDO.

Table 3-35. 0x32 TX_SOURCE_CAPS Register Bit Field Definitions (continued)

Bits	Name	Description	
1:0	PowerPathForPDO1	Configures which PP to use for PDO1.	
		00b	PP_5V1 is used for this PDO.
		01b	PP_5V2 is used for this PDO.
		10b	PP_EXT1 is used for this PDO.
		11b	PP_EXT2 is used for this PDO.
Byte 1: Header			
7:3	Reserved		
2:0	numValidPDOs	Number of valid PDOs in this register. Each PDO is 4 bytes. (max of 7)	

The PDO's in this register follow the definition in the USB PD specification. It is reproduced here for convenience, but for more details on each field refer to the USB PD specification.

Table 3-36. First PDO

Bits(s)	Description
31:30	Supply Type, this shall always be set to 00b (Fixed Supply)
29	Dual-Role Power, this is overridden by the logical OR of the ProcessSwapToSink, ProcessSwapToSource, InitiateSwapToSink, and InitiateSwapToSource fields in the PORT_CONTRL register.
28	USB Suspend Supported
27	Unconstrained Power, this is overridden by PORT_CONTROL.UnconstrainedPower.
26	USB Communiations Capable, this is overridden by PORT_CONFIG.UsbCommCapable.
25	Dual-Role Data, this is overridden by the logical OR of the ProcessSwapToUFP, ProcessSwapToDFP, InitiateSwapToUFP, and InitiateSwapToDFP fields in the PORT_CONTRL register.
24	Unchunked Extended Messages supported.
23:22	Reserved
21:20	Peak Current
19:10	Voltage
9:0	Maximum Current

Table 3-37. Other PDO's.

Bits(s)	Description		
	Fixed Supply	Variable Supply	Battery Supply
31:30	00b	01b	10b
29:20	Reserved.	Maximum Voltage	Maximum Voltage
19:10	Voltage	Minimum Voltage	Minimum Voltage
9:0	Maximum Current	Maximum Current	Maximum Allowable Power

3.18 0x33 TX_SINK_CAPS Register

The PD controller transmits the contents of this register as a Sink_Capabilities message after receiving a Get_Sink_Cap message unless its configuration or USB PD rules require a different response in the context.

NOTE: Writes to this register have no immediate effect. The PD controller updates and uses this register each time it needs to send a *Sink Capabilities* message.

Table 3-38. 0x33 TX_SINK_CAPS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x33	TX_SINK_CAPS	RW	29	yes	Initialized by Application Configuration

Table 3-39. 0x33 TX_SINK_CAPS Register Bit Field Definitions

Bits	Name	Description
Bytes 26-29: PDO #7 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO7	Seventh Sink Capabilities PDO contents. See Table 3-41
Bytes 22-25: PDO #6 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO6	Sixth Sink Capabilities PDO contents. See Table 3-41
Bytes 18-21: PDO #5 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO5	Fifth Sink Capabilities PDO contents. See Table 3-41
Bytes 14-17: PDO #4 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO4	Fourth Sink Capabilities PDO contents. See Table 3-41
Bytes 10-13: PDO #3 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO3	Third Sink Capabilities PDO contents. See Table 3-41
Bytes 6-9: PDO #2 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO2	Second Sink Capabilities PDO contents. See Table 3-41
Bytes 2-5: PDO #1 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO1	First Sink Capabilities PDO contents. See Table 3-40
Byte 1: Header		
7:3	Reserved	
2:0	numValidPDOs	

Each PDO in this TX_SINK_CAPS register follows the definition from the USB PD specification, reproduced below for convenience. For more details on the meaning of each field refer to the USB PD specification.

Table 3-40. First PDO

Bits(s)	Description
31:30	Supply Type, this shall always be set to 00b (Fixed Supply)
29	Dual-Role Power, this is overridden by the logical OR of the ProcessSwapToSink, ProcessSwapToSource, InitiateSwapToSink, and InitiateSwapToSource fields in the PORT_CONTRL register.
28	Higher Capability
27	Unconstrained Power, this is overridden by PORT_CONTROL.UnconstrainedPower.
26	USB Communiations Capable, this is overridden by PORT_CONFIG.UsbCommCapable.
25	Dual-Role Data, this is overridden by the logical OR of the ProcessSwapToUFP, ProcessSwapToDFP, InitiateSwapToUFP, and InitiateSwapToDFP fields in the PORT_CONTRL register.
24:23	Fast-Role Swap Required Current. Not all PD controllers support Fast-Role swap as initial source. For those PD controllers this field should always be 00b.
22:20	Reserved

Table 3-40. First PDO (continued)

Bits(s)	Description
19:10	Voltage
9:0	Operational Current

Table 3-41. Other PDO's.

Bits(s)	Description			
	Fixed Supply	Variable Supply	Battery Supply	APDO (PPS)
31:30	00b	01b	10b	11b
29:28	Reserved.	Maximum Voltage	Maximum Voltage	00b
27:25				Reserved
24:20				MaxPpsVoltage
19:17	Voltage	Minimum Voltage	Minimum Voltage	
16				Reserved
15:10				MinPpsVoltage
9:8	Operational Current	Operational Current	Operational Power	
7				Reserved
6:0				MaxPpsCurrent

3.19 0x34 ACTIVE_CONTRACT_PDO Register

Table 3-42. 0x34 ACTIVE_CONTRACT_PDO Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x34	ACTIVE_CONTRACT_PDO	RO	6	yes	Cleared on disconnect, Hard Reset, or PR_Swap.

Table 3-43. 0x34 ACTIVE_CONTRACT_PDO Register Bit Field Definitions

Bits	Name	Description
Bytes 5-6: Source Properties		
15:10	Reserved	
9:0	firstPDOControlBits	Contains bits 29:20 of the first PDO. It does not matter which PDO was selected, this field is always drawn from the first PDO.
Bytes 1-4: Contract PDO (treated as 32-bit little endian value)		
31:0	ActivePDO	Power data object. This field contains the contents of the PDO Requested by PD Controller as Sink and Accepted by Source, once it is Accepted by Source.

3.20 0x35 ACTIVE_CONTRACT_RDO Register

Table 3-44. 0x35 ACTIVE_CONTRACT_RDO Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x35	ACTIVE_CONTRACT_RDO	RO	4	yes	Cleared on disconnect, Hard Reset, or PR_Swap.

Table 3-45. 0x35 ACTIVE_CONTRACT_RDO Register Bit Field Definitions

Bits	Name	Description
Bytes 1-4: Contract RDO (treated as 32-bit little endian value)		
31	Reserved	
30:28	ObjectPosition	As defined by USB PD.
27	GiveBackFlag	As defined by USB PD.
26	CapabilityMismatch	As defined by USB PD.
25	USBCommCapable	As defined by USB PD.
24	NoUSBSuspend	As defined by USB PD.
23	UnchunkedSupported	As defined by USB PD.
22:20	Reserved	
19:10	OperatingX	As defined by USB PD.
9:0	MaxMinOperatingX	As defined by USB PD.

3.21 0x37 AUTO_NEGOTIATE_SINK Register

This register allows the host to configure how the PD controller responds to the last Source Capabilities messages it received.

In general, writing this register while a sink contract is in place will not cause an automatic renegotiation, changes will take effect the next time a contract is negotiated. The *ANeg* command forces a re-evaluation of this register and a new *Request* message will be issued if appropriate.

If the first four bytes of this register are written as zero, then the PD controller will always request a 5V Fixed Suppy contract at 100 mA.

Below is a high-level summary of how this register drives the PDO selection.

- Parse the received PDO's in the register RX_SOURCE_CAPS. Discard any PDO whose voltage range is below ANMinVoltage or above ANMaxVoltage.
- Calculate the PDO power for each received PDO (RX_SOURCE_CAPS.SourcePdoX). Rank all PDO's according to the PDO power.
  - PDO Power = Voltage * MaximumCurrent (Fixed Supply)
  - PDO Power = MinimumVoltage * MaximumCurrent (Variable Supply)
  - PDO Power = MaximumPower (Battery Supply)
- The PDO with maximum PDO Power that also passes the voltage check is selected. In case there are multiple PDO's that pass the voltage check and have the same maximum PDO Power, several tie breakers are applied as described below:
  - A Fixed supply type is preferred, and Variable supply type is preferred over Battery supply type.
  - If the PDO's being compared have the same supply type, then ANRDOPriority specifies how to break the tie.

Table 3-46. 0x37 AUTO_NEGOTIATE_SINK Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x37	AUTO_NEGOTIATE_SINK	RW	16	yes	Initialized by Application Configuration

Table 3-47. 0x37 AUTO_NEGOTIATE_SINK Register Bit Field Definitions

Bits	Name	Description
Bytes 2-8: Auto-Negotiate Requirements		
55:54	Reserved	
53:44	ANSinkCapMismatchPower	Capabilities Mismatch Power Threshold. If the selected PDO offers less power than what is specified in this register, then the PD controller will assert the Capability Mismatch bit in its Request message unless NoCapabilityMismatch is set to 1. (250mW per LSB)
43:34	ANMinVoltage	MInimum voltage to request. During PD power contract negotiation, the PD controller will only select voltages that are greater than or equal to the value specified in this field. Not used unless AutoComputeSinkMinVoltage=0. (50mV per LSB)
33:24	ANMaxVoltage	Maximum voltage to request. During PD power contract negotiation, the PD controller will only select voltages that are less than or equal to the value specified in this field. Not used unless AutoComputeSinkMinVoltage=0. (50mV per LSB)
23:14	ANSinkMinRequiredPower	Minimum operating power required by the Sink. The PD Controller will always attempt to receive this power level from the Source. (250mW per LSB) See description in AutoComputeSinkMinPower field
13:4	ANMaxCurrent	Maximum current to request. The PD controller will not request more current than indicated by this field. The host should ensure that the max current for all PDO's in the TX_SINK_CAPS register do not exceed this value. (10mA per LSB).
3:0	Reserved	

Table 3-47. 0x37 AUTO_NEGOTIATE_SINK Register Bit Field Definitions (continued)

Bits	Name	Description	
Byte 1: Auto Negotiate Configuration			
7	Reserved		
6	AutoDisableSinkUponCapMismatch	Sink path and capability mismatch settings. If this bit is asserted, then any time the implicit or explicit power contract would cause the Capability Mismatch bit to be set the PD controller will disable the sinking path. The 'SRDY' 4CC task can override and enable the sink path. However, if the contract changes after the the 'SRDY' has completed, the PD controller will disable the sink path if the contract causes a capability mismatch. This bit should only be asserted if the NoCapabilityMismatch bit is set to 0.	
5	AutoComputeSinkMaxVoltage	Configuration for maximum voltage. The PD controller can automatically compute ANMaxVoltage, or allow the host to specify it.	
		0b	Provided by host. The PD controller will not automatically compute ANMaxVoltage. The host is required to write a valid value in the ANMaxVoltage field either during configuration or dynamically.
		1b	Computed by PD controller. The PD controller automatically sets ANMaxVoltage to the maximum voltage from the valid PDO's in TX_SINK_CAPS register (0x33).
4	AutoComputeSinkMinVoltage	Configuration for minimum voltage. The PD controller can automatically compute ANMinVoltage, or allow the host to specify it.	
		0b	Provided by host. The PD controller will not automatically compute ANMinVoltage. The host is required to write a valid value in the ANMinVoltage field either during configuration or dynamically.
		1b	Computed by PD controller. The PD controller automatically sets ANMinVoltage field to 4.75V.
3	NoCapabilityMismatch	Configuration for capability mismatch in RDO. There are two conditions that will trigger a capability mismatch: • If the attached source does not offer a PDO whose power is greater or equal to the ANSinkCapMismatchPower field in this register. • PPS is enabled in this register and the attached source did not offer a PPS PDO that matches the requirements in TX_SINK_CAPS. If either condition is true, then the PD controller will assert the capability mismatch bit in its request unless this bit is asserted.	
		0b	Capabiltiy mismatch enabled. The capability mismatch bit will be asserted to indicate to the attached source that more power is desired.
		1b	Capability mismatch disabled. The capability mismatch bit will not be asserted.
2	AutoComputeSinkMinPower	Minimum power sink requires. The minimum sink power is the largest power reported in any valid PDO in the TX_SINK_CAPS (0x33). The power for a particular PDO from the TX_SINK_CAPS follows for each supply type: • *Battery Supply*: OperatingPower • *Variable Supply*: MaxVoltage*OperatingCurrent • *Fixed Supply*: Voltage*OperatingCurrent. However, if the TX_SINK_CAPS register includes Battery supply type PDO(s), then ANSinkMinRequiredPower = maximum OperatingPower in a Battery supply type PDO.	
		0b	Provided by host. The host is required to write the ANSinkMinRequiredPower field of this register.
		1b	Computed by PD controller. The PD controller will compute the ANSinkMinRequiredPower field in this register, the host need not write that field when writing this register. If the host does write the ANSinkMinRequiredPower field, the PD controller will overwrite it with its own calculation.
1	NoUSBSusp	Value used for the NoUSBSusp Flag in the RDO. This is as defined by USB PD.	

Table 3-47. 0x37 AUTO_NEGOTIATE_SINK Register Bit Field Definitions (continued)

Bits	Name	Description	
0	ANRDOPriority	Configuration for tie-breaker in PDO selection. The PD controller will find the set of PDO's that fulfill the voltage requirements. From that set of PDO's it will pick the one with higher power. If two acceptable PDO's have the same power, Fixed Supply Type is preferred, and then Variable Supply has second preference. If two PDO's have the same power and the same type, then this bit determines which PDO is selected.	
		0b	Higher voltage. • For Fixed Supply Type: select the PDO with higher voltage. • For Variable and Battery Supply Types: select the PDO with the highest MinVoltage
		1b	Lower voltage. • For Fixed Supply Type: select the PDO with lower voltage. • For Variable and Battery Supply Types: select the PDO with the lowest MaxVoltage

3.21.1 AUTO_NEGOTIATE_SINK usage example #1

When attached to a 36W source the PD controller has RX_SOURCE_CAPS:

- PDO1: 5V @ 3A
- PDO2: 9V @ 3A
- PDO3: 15V @ 2.4A
- PDO4: 20V @ 1.8A

The PD controller has TX_SINK_CAPS set as:

- PDO1: 5V @ 3A (fixed)
- PDO2: 20V @ 3A (fixed)

The PD controller has AUTO_NEGOTIATE_SINK set as:

- AUTO_NEGOTIATE_SINK = 0
- AUTO_NEGOTIATE_SINK.ANSinkCapMismatchPower = 240d (60W)
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMinPower = 1
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMinVoltage = 1
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMaxVoltage = 1
- AUTO_NEGOTIATE_SINK.NoCapabilityMismatch = x (see table below)
- AUTO_NEGOTIATE_SINK.ANRDOPriority = y (see table below)

The settings give the following results:

- ANSinkMinRequiredPower computed as 60 W
- ANMaxVoltage computed as 20V
- ANMinVoltage computed as 4.75V

Table 3-48. AUTO_NEGOTIATE_SINK usage example #1.

AUTO_NEGOTIATE_SINK		ACTIVE_CONTRACT_RDO			
NoCapabilityMismatch	ANRDOPriority	OperatingX	MinMaxOperatingX	ObjectPosition	Capability Mismatch
0	0	1.8A	3.0A	4	1
1	0	1.8A	1.8A	4	0
1	1	2.4A	2.4A	3	0

3.21.2 AUTO_NEGOTIATE_SINK usage example #2

When attached to a 36W source the PD controller has RX_SOURCE_CAPS:

- PDO1: 5V @ 3A
- PDO2: 9V @ 3A
- PDO3: 15V @ 2.4A

The PD controller has TX_SINK_CAPS set as:

- PDO1: 5V @ 0.1A (fixed)
- PDO2: 20V @ 3A (fixed)

The PD controller has AUTO_NEGOTIATE_SINK set as:

- AUTO_NEGOTIATE_SINK = 0
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMinPower = 1
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMinVoltage = 0
- AUTO_NEGOTIATE_SINK.ANMinVoltage = 20V
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMaxVoltage = 1
- AUTO_NEGOTIATE_SINK.NoCapabilityMismatch = x (see table below)
- AUTO_NEGOTIATE_SINK.ANRDOPriority = 0

The settings give the results in the table below. Note that ANMaxVoltage computed as 20V, but it doesn't affect the result. Since the ANMinVoltage was set to 20V, and the source is not offering 20V none of the source PDO's fulfill the sink requirements. Even though ANSinkCapMismatchPower=0 in this example, since the voltages offered are insufficient, the capability mismatch bit may still be set.

Table 3-49. AUTO_NEGOTIATE_SINK usage example #2.

AUTO_NEGOTIATE_SINK	ACTIVE_CONTRACT_RDO			
NoCapabilityMismatch	**OperatingX**	**MinMaxOperatingX**	**ObjectPosition**	**Capability Mismatch**
0	3.0A	3.0A	1	1
1	3.0A	3.0A	1	0

3.21.3 AUTO_NEGOTIATE_SINK usage example #3

When attached to a 45W source the PD controller has RX_SOURCE_CAPS:

- PDO1: 5V @ 3A
- PDO2: 9V @ 3A
- PDO3: 15V @ 3A
- PDO4: 20V @ 2.25A

The PD controller has TX_SINK_CAPS set as:

- PDO1: 5V @ 3A (fixed)
- PDO2: 20V @ 2.25A (fixed)

The PD controller has AUTO_NEGOTIATE_SINK set as:

- AUTO_NEGOTIATE_SINK = 0
- AUTO_NEGOTIATE_SINK.ANSinkCapMismatchPower = 180d (45W)
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMinPower = 1
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMinVoltage = 1
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMaxVoltage = 1
- AUTO_NEGOTIATE_SINK.NoCapabilityMismatch = 0
- AUTO_NEGOTIATE_SINK.ANRDOPriority = y (see table below)

The settings give the following results:

- ANSinkMinRequiredPower computed as 45 W
- ANMaxVoltage computed as 20V
- ANMinVoltage computed as 4.75V

Table 3-50. AUTO_NEGOTIATE_SINK usage example #3.

AUTO_NEGOTIATE_SINK	ACTIVE_CONTRACT_RDO			
ANRDOPriority	OperatingX	MinMaxOperatingX	ObjectPosition	Capability Mismatch
0	2.25A	2.25A	4	0
1	3.0A	3.0A	3	0

3.21.4 AUTO_NEGOTIATE_SINK usage example #4

When attached to a 100W source the PD controller has RX_SOURCE_CAPS:

- PDO1: 5V @ 3A
- PDO2: 9V @ 3A
- PDO3: 15V @ 3A
- PDO4: 20V @ 5A

The PD controller has TX_SINK_CAPS set as:

- PDO1: 5V @ 3A (fixed)
- PDO2: 20V @ 5A (fixed)

The PD controller has AUTO_NEGOTIATE_SINK set as:

- AUTO_NEGOTIATE_SINK = 0
- AUTO_NEGOTIATE_SINK.ANSinkCapMismatchPower = 240d (60W)
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMinPower = 1
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMinVoltage = 1
- AUTO_NEGOTIATE_SINK.AutoComputeSinkMaxVoltage = 1
- AUTO_NEGOTIATE_SINK.NoCapabilityMismatch = 0
- AUTO_NEGOTIATE_SINK.ANRDOPriority = y (see table below)

The settings give the following results:

- ANSinkMinRequiredPower computed as 100 W
- ANMaxVoltage computed as 20V
- ANMinVoltage computed as 4.75V

Table 3-51. AUTO_NEGOTIATE_SINK usage example #3.

AUTO_NEGOTIATE_SINK	ACTIVE_CONTRACT_RDO			
ANRDOPriority	OperatingX	MinMaxOperatingX	ObjectPosition	Capability Mismatch
0	5A	5A	4	0
1	5A	5A	4	0

3.22 0x3F POWER_STATUS Register

Table 3-52. 0x3F POWER_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x3F	POWER_STATUS	RO	2	yes	Cleared on disconnect.

Table 3-53. 0x3F POWER_STATUS Register Bit Field Definitions

Bits	Name	Description	
15:10	Reserved	Reserved	
9:8	ChargerAdvertiseStatus[1][2]	Charger Advertise Status	
		00b	Charger advertise disabled or not run
		01b	Charger advertisement in process
		10b	Charger advertisement complete
		11b	Reserved
7:4	ChargerDetectStatus[1][2]		
		0h	Charger detection disabled or not run
		1h	Charger detection in progress
		2h	Charger detection complete, none detected
		3h	Charger detection complete, SDP detected
		4h	Charger detection complete, BC 1.2 CDP detected
		5h	Charger detection complete, BC 1.2 DCP detected
		6h	Charger detection complete, Divider1 DCP detected
		7h	Charger detection complete, Divider2 DCP detected
		8h	Charger detection complete, Divider3 DCP detected
		9h	Charger detection complete, 1.2V DCP detected
		Ah-Fh	Reserved
3:2	TypeCCurrent	This field is redundant with PD_STATUS.CCPullUp in register 0x40 when SourceSink is 1b. This field is redundant with PORT_CONTROL.TypeCCurrent in register 0x29 when SourceSink is 0b. In the future, this redundant field may be removed. This field is intended for Type-C Sink operation. If the port is connected as source, the field is updated upon initial connection only.	
		00b	USB Default Current
		01b	1.5 A
		10b	3.0 A
		11b	Explicit PD contract sets current. A PD contract was negotiated (see other PD registers for more details).
1	SourceSink	Source / Sink indicator. This bit is equivalent to PresentPDRole in register 0x40. It is replicated in this register for convenience. In the future, this redundant bit may be removed.	
		0b	Connection requests power. The PD Controller is the source.
		1b	Connection provides power (PD Controller as sink).
0	PowerConnection	Asserted if there is a connection. This bit is asserted when PlugPresent is TRUE and ConnState is greater than 5h. So it is redundant with information from register 0x1A. It is replicated in this register for convenience. In the future this redundant bit may be removed.	
		0b	No connection. The rest of bits in this register are not valid.
		1b	Connection present. See other bits in register for more details.

[1] This feature is not supported by TPS65994AD_F509.04.02.

[2] This feature is not supported by TPS65993AD_F509.04.02.

3.23 0x40 PD_STATUS Register

Table 3-54. 0x40 PD_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x40	PD_STATUS	RO	4	yes	Cleared on connect.

Table 3-55. 0x40 PD_STATUS Register Bit Field Definitions

Bits	Name	Description	
31	Reserved	Reserved	
30:28	DataResetDetails	Reason for Data Reset.	
		0h	Reset value: no data reset.
		1h	Data Reset message received from port partner.
		2h	Requested by host: 'DRST'
		3h	Requested by host: DATA_CONTROL
		4h	Exit USB4 following DR_Swap
		5h	Reserved1
		6h	Reserved2
		7h	Reserved3

Table 3-55. 0x40 PD_STATUS Register Bit Field Definitions (continued)

Bits	Name	Description	
27:22	ErrorRecoveryDetails	Reason for Error Recovery	
		00h	reset value: no error recovery
		01h	System: over-temperature shut-down.
		02h	System: PP5V went low unexpectedly.
		03h	System: fault input GPIO was asserted. GPIO could be either Fault_Input_Event_Port1 or Fault_Input_Event_Port2.
		04h	System: Over-voltage detected on the Px_VBUS pin.
		05h	Reserved
		06h	System: ILIM on PP_5V
		07h	System: ILIM on PP_CABLE
		08h	System: OVP on CC detected.
		9h-Fh	Reserved
		10h	Protocol error: invalid DR_Swap.
		11h	Protocol error: no Good_CRC during a PR_Swap sequence. This happens if the sink did not turn off in time.
		12h	Protocol error: no Good_CRC during a FR_Swap sequence. This happens if the source did not turn off in time.
		13h-14h	Reserved
		15h	Policy Engine: NoResponse timer timed out.
		16h	Policy Engine: PSSourceOffTimer timed out during PR_Swap.
		17h	Policy Engine: PSSourceOnTimer timed out during PR_Swap.
		18h	Policy Engine: PSSourceOnTimer timed out during FR_Swap.
		19h	Policy Engine: The Type-C source failed to change during FR_Swap.
		1Ah	Policy Engine: SenderResponseTimer timed out during FR_Swap.
		1Bh	Policy Engine: PSSourceOffTimer timed out during FR_Swap.
		1Ch	Policy Engine: Error reaching the Attached.SNK state during dead-battery due to changing which PP_EXTx path is enabled.
		1Dh-1Fh	Reserved
		20h	HI: PortConfig.TypeCStateMachine is set to disabled.
		21h	HI: Error with DATA_CONTROL. The HostConnected bit in the DATA_CONTROL register (0x50) changed and the resulting configuration was not compatible with the existing state of the port.
		22h	HI: Swapping error during dead-battery. After Host cleared the dead-battery flag and the configuration requires operating as a source, but the PD controller is unable to swap to sink.
		23h	HI: Host updated the GLOBAL_SYSTEM_CONFIG register (0x27).
		24h	HI: Host issued the 4CC 'GAID' command.
		25h	HI: Host issued the 4CC 'Gaid' command.
		26h	HI: Host issued the 4CC 'DISC' command.
		27h	HI: Host issued a reset using the 'UCSI' 4CC command.
		28h-2Fh	Reserved
		30h	Type-C: an error occurred in the Attached.SRC state.
		31h	Type-C: VCONN failed to discharge.
		32h-3Fh	Reserved

Table 3-55. 0x40 PD_STATUS Register Bit Field Definitions (continued)

Bits	Name	Description	
21:16	HardResetDetails	Reason for Hard Reset	
		00h	Reset value, no hard reset.
		01h	Received from Port Partner.
		02h	Requested by host.
		03h	Invalid DR_Swap request during Active Mode
		04h	Required by policy engine: DischargeFailed.
		05h	Required by policy engine: NoResponseTimeOut.
		06h	Required by policy engine: SendSoftReset.
		07h	Required by policy engine: Sink_SelectCapability.
		08h	Required by policy engine: Sink_TransitionSink.
		09h	Required by policy engine: Sink_WaitForCapabilities.
		0Ah	Required by policy engine: SoftReset.
		0Bh	Required by policy engine: SourceOnTimeout.
		0Ch	Required by policy engine: Source_CapabilityResponse.
		0Dh	Required by policy engine: Source_SendCapabilities.
		0Eh	Required by policy engine: SourcingFault.
		0Fh	Required by policy engine: UnableToSource.
		10h	Required by policy engine: FRS failure
		11h	Required by policy engine: Unexpected message
		12h	Required by policy engine: Failure to to complete the VCONN recovery sequence within 200ms after PP5V rising edge.
		13h-3Fh	Reserved.
15:13	Reserved		

Table 3-55. 0x40 PD_STATUS Register Bit Field Definitions (continued)

Bits	Name	Description	
12:8	SoftResetDetails	Reason for Soft Reset	
		0h	Reset value, no soft reset.
		1h	Soft reset received from Port Partner.
		2h	Reserved
		3h	Reserved
		4h	Received source capabilities message was invalid.
		5h	Message retries were exhausted.
		6h	Received an accept message unexpectedly.
		7h	Received a control message unexpectedly.
		8h	Received a GetSinkCap message unexpectedly.
		9h	Received a GetSourceCap message unexpectedly.
		Ah	Received a GotoMin message unexpectedly.
		Bh	Received a PS_RDY message unexpectedly.
		Ch	Received a Ping message unexpectedly.
		Dh	Received a Reject message unexpectedly.
		Eh	Received a Request message unexpectedly.
		Fh	Received a Sink Capabilities message unexpectedly.
		10h	Received Source Capabilities message unexpected.
		11h	Received a Swap message unexpectedly.
		12h	Received a Wait Capabilities message unexpectedly.
		13h	Received an unknown control message.
		14h	Received an unknown data message.
		15h	To initialize SOP' controller in plug
		16h	To initialize SOP" controller in plug
		17h	Received an Extended message unexpectedly.
		18h	Received an unknown Extended message.
		19h	Received a data message unexpectedly.
		1Ah	Received a Not Supported message unexpectedly.
		1Bh	Received a Get_Status message unexpectedly.
		1Ch-1Fh	Reserved
7	Reserved		
6	PresentPDRole	Present PD power role. The PD Controller is acting under this PD power role.	
		0b	Sink
		1b	Source
5:4	PortType	Present Type-C power role. The PD Controller is acting under this Type-C power role.	
		00b	Sink/Source
		01b	Sink
		10b	Source
		11b	Source/Sink
3:2	CCPullUp	CC Pull-up value. The pull-up value detected by PD Controller when in CC Pull-down mode.	
		00b	Not in CC pull-down mode / no CC pull-up detected.
		01b	USB Default current
		10b	1.5 A (SinkTxNG)
		11b	3.0 A (SinkTxOK)
1:0	Reserved		

3.24 0x41 PD3_STATUS Register

Table 3-56. 0x41 PD3_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x41	PD3_STATUS	RO	4	yes	Cleared on connect.

Table 3-57. 0x41 PD3_STATUS Register Bit Field Definitions

Bits	Name	Description	
Bytes 3-4: Negotiated Spec Revisions			
15	useUnchunkedMessages	Unchunking support (SOP). Indicates if Port Partner supports unchunked messages.	
14:13	plugNegotiatedSpecRev	USB PD specification revision (SOP'). As negotiated with Cable Plug.	
		00b	Revision 1
		01b	Revision 2
		10b	Revision 3
		11b	Reserved
12:11	portNegotiatedSpecRev	USB PD specification revision (SOP). As negotiated with Port Partner.	
		00b	Revision 1
		01b	Revision 2
		10b	Revision 3
		11b	Reserved
10:9	plugNegotiatedSvdmVer	SVDM specification version (SOP'). As negotiated with Cable Plug.	
		00b	Version 1
		01b	Version 2
		10b	Version 3
		11b	Reserved
8:7	portNegotiatedSvdmVer	SVDM specification version (SOP). As negotiated with Port Partner.	
		00b	Version 1
		01b	Version 2
		10b	Version 3
		11b	Reserved
6:0	Reserved		
Byte 2: Reserved			
Byte 1: Reserved			

3.25 0x42 PD3_CONFIG Register

Table 3-58. 0x42 PD3_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x42	PD3_CONFIG	RW	3	no	Initialized by Application Configuration

Table 3-59. 0x42 PD3_CONFIG Register Bit Field Definitions

Bits	Name	Description
23:18	Reserved	
17	SupportSinkCapExtended	Support Sink Capabilities Extended message. If this bit is asserted the PD controller will respond to a Get_Sink_Capabilities_Extended message USB PD message with the contents of the TX_SKEDB register (0x7E).
16	SupportCountryCodeInfo	Support Country Code and information messages. If this bit is asserted the PD controller will use the data stored during application configuration to respond to a Get_Country_Code and Get_Country_Info message. See also BINARYDATA_INDICES register (0x62).
15	Reserved	
14	Reserved	
12	SupportManufactureInfoMsg	Support Manufacturing Info message. If this bit is asserted the PD controller will respond to a Get_Manufacturer_Info USB PD message with the contents of the TX_MIDB_SOP register (0x73).
11	SupportBatteryStatusMsg	Support Battery Status message. If this bit is asserted the PD controller will respond to a Get_Battery_Status USB PD message with the contents of the TX_BSDO register (0x7B).
10	SupportBatteryCapMsg	Support Battery Capability message. If this bit is asserted the PD controller will respond to a Get_Battery_Capabilities USB PD message with the contents of the TX_BCDB register (0x7D).
9	SupportStatusMsg	Enable Status message. If this bit is asserted the PD controller will respond to a Get_Status USB PD message with the contents of the TX_SDB register (0x79).
8	SupportSouceCapExtMsg	Enable Source Capabilities Extended. If this bit is asserted the PD controller will respond to a Get_Source_Capabilities_Extended USB PD message with the contents of the TX_SCEDB register (0x77).
7	Reserved	
6	Reserved	
5	Reserved	
4	unchunkedSupported	Enable unchunked support. If this bit is asserted the PD controller will support unchunked messaging (up to 260 bytes). The host is responsible to consume the unchunked message before the PD controller will be able to receive another long unchunked message.
3:2	plugMaxSpecRev	Cable Plug's highest USB PD spec revision. 00b Reserved 01b Revision 2 (USB PD 2.0) 10b Revision 3 (USB PD 3.0) 11b Reserved
1:0	portMaxSpecRev	Port Partner's highest USB PD spec revision. 00b Reserved 01b Revision 2 (USB PD 2.0) 10b Revision 3 (USB PD 3.0) 11b Reserved

3.26 0x43 DELAY_CONFIG Register

Table 3-60. 0x43 DELAY_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x43	DELAY_CONFIG	RW	28	no	Initialized by Application Configuration

Table 3-61. 0x43 DELAY_CONFIG Register Bit Field Definitions

Bits	Name	Description
Bytes 25-28: Seventh Set of Delays		
Bytes 21-24: Sixth Set of Delays		
Bytes 17-20: Fifth Set of Delays		
Bytes 13-16: Fourth Set of Delays		
Bytes 9-12: Third Set of Delays		
31:24	tDataStatusNotificationDelay	The PD controller will ensure that INT_EVENTx.DataStatusUpdate will not get asserted again unless a minimum time has elapsed since the last time it was asserted. This register specifies that minimum time (500us per LSB).
23:16	tDisconnectDelay	Disconnect delay. After a disconnect event, the PD controller will delay for the amount of time specified in this register before asserting Retimer_Reset_N_Event_Portx to reset the retimer. (1ms per LSB)
15:12	Reserved	
11:8	ResetzBootDelay	RESETZ boot delay. The RESETZ GPIO Event will be asserted for this long when the PD controller enters into 'APP ' mode (50ms per LSB).
7:4	MresetDelayFall	Delay for MRESET (falling). Configurable delay for the MRESET and RESETZ GPIO Events (50ms per LSB).
3:0	MresetDelayRise	Delay for MRESET (rising). Configurable delay for the MRESET and RESETZ GPIO Events (50ms per LSB).
Bytes 5-8: Second Set of Delays		
31:24	RetimerForcePwrDelayFall	Retimer force power delay (falling). Configurable delay between Retimer_Reset_N GPIO falling and Retimer_Force_PWR_Event GPIO falling. If this field is set to 0x00, then the falling time will be the same as RetimerForcePwrDelayRise. (10ms per LSB)
23:16	RetimerForcePwrDelayRise	Retimer force power delay (rising). Configurable delay between Retimer_Force_PWR_Event GPIO rising and Retimer_Reset_N GPIO rising (250us per LSB).
15:8	ConnectionDelay	Connection delay. Configurable delay from initial connection configuration to start of PD negotiation (500us per LSB).
7:0	MuxDelay_UFP_config	Mux delay for UFP configuration. Configurable delay from entering configuration to ACK response (500us per LSB).
Bytes 1-4: First Set of Delays		
31:24	HpdDelay	Mux delay for HPD change. Configurable delay after changing HPD state (500us per LSB).
23:16	MuxDelay_DRS	Mux delay for DR_Swap. Configurable delay after changing mux data role (500us per LSB).
15:8	MuxDelay_UFP	Mux delay for UFP. Configurable delay from entering mode to ACK response (500us per LSB).
7:0	MuxDelay_DFP	Mux delay for DFP. Configurable delay from entering safe state to sending mode enter command (500 us per LSB)

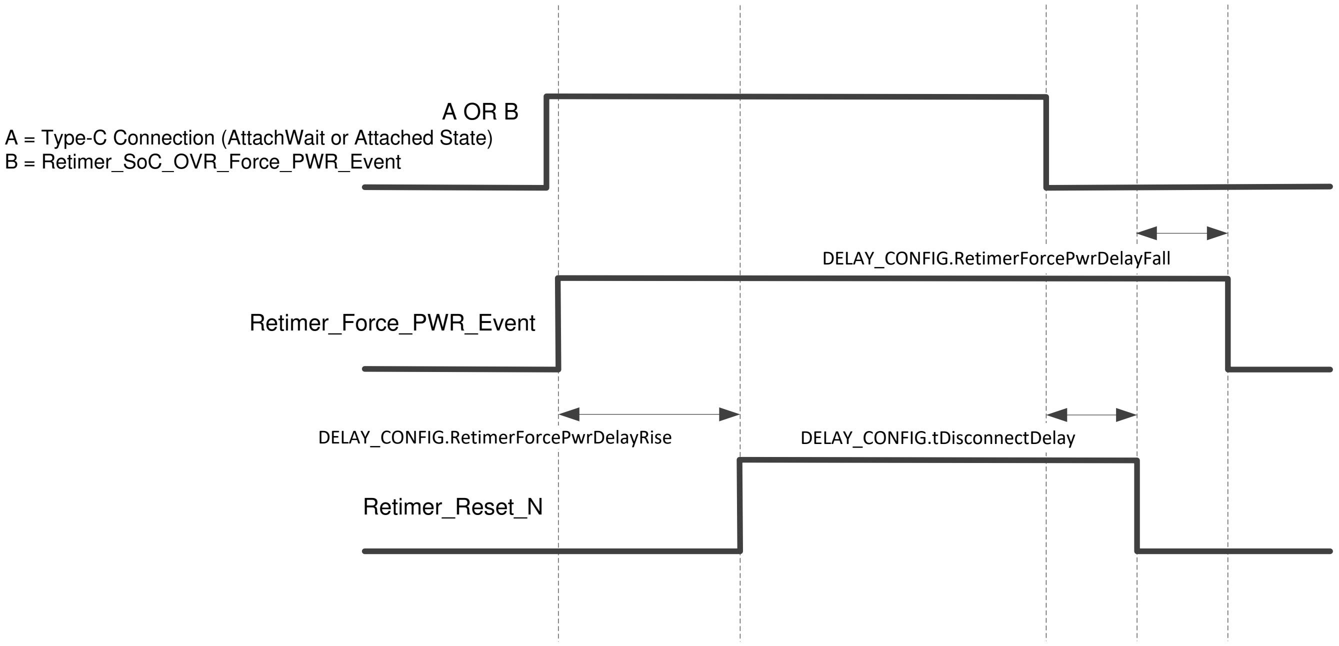

A OR B
A = Type-C Connection (AttachWait or Attached State)
B = Retimer_SoC_OVR_Force_PWR_Event
DELAY_CONFIG.RetimerForcePwrDelayFall
Retimer_Force_PWR_Event
DELAY_CONFIG.RetimerForcePwrDelayRise
DELAY_CONFIG.tDisconnectDelay
Retimer_Reset_N

Figure 3-1. Timing diagram for Retimer_Force_PWR_Event and Retimer_Reset_N.

3.27 0x47 TX_IDENTITY Register

The PD controllers transmits the contents of this register as a Discover Identity ACK message after receiving a Discover Identity REQ message. The PD controller is not designed to transmit Product Type VDO as part of this ACK message, that is only required for Alternate Mode Adaptors (AMA), VCONN Powered Devices (VPD), and cables.

NOTE: Writes to this register have no immediate effect. PD Controller will update and use the contents of this register each time a Discover Identity SVDM is received before generating the Discover Identity SVDM ACK message.

Table 3-62. 0x47 TX_IDENTITY Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x47	TX_IDENTITY	RW	25	yes	Initialized by Application Configuration

Table 3-63. 0x47 TX_IDENTITY Register Bit Field Definitions

Bits	Name	Description
Bytes 22-25: VDO #6		
31:0	ProductTypeVdo2SOP	
Bytes 18-21: VDO #5		
31:0	ProductTypeVdo1SOP	
Bytes 14-17: VDO #4		
31:0	ProductTypeVdo0SOP	
Bytes 10-13: VDO #3: Product VDO		
31:16	ProductId	Product ID. The value written for PID here is used to populate the PID in other registers as well: TX_MIDB_SOP, TX_SCEDB, and TX_SKEDB.
15:0	bcdDevice	This field is read-only and contains the FW version for the PD controller.
Bytes 6-9: VDO #2 (treated as a 32-bit little endian value)		
31:0	CertStatVdo	This contains the 32-bit XID assigned by USB-IF. The value written for XID here is used to populate the XID in other registers as well: TX_SCEDB, and TX_SKEDB.
Bytes 2-5: VDO #1: ID Header VDO		
31	USBCommCapableAsHost	Assert if USB communications capable as a host.
30	USBCommCapableAsDevice	Assert if USB communications capable as a device.
29:27	ProductType_UFP	This is as defined in the USB PD specification. bits 15:0 are the VID. The value written for VID here is used to populate the VID in other registers as well: TX_MIDB_SOP, TX_SCEDB, and TX_SKEDB. 000b Undefined 001b PDUSB Hub 010b PDUSB peripheral 011b Power Sinking Device (PSD) 100b Reserved 101b Alternate Mode Adaptor (AMA) 110b Vconn-Powered device (VPD) 111b Reserved
26	ModalOperationSupported	Assert this bit if Alternate Modes are supported.

Table 3-63. 0x47 TX_IDENTITY Register Bit Field Definitions (continued)

Bits	Name	Description	
25:23	ProductType_DFP	This is as defined in the USB PD specification.	
		000b	Undefined
		001b	PDUSB Hub
		010b	PDUSB Host
		011b	Power brick
		100b	Alternate Mode Controller (AMC)
		101b	Reserved
		110b	Reserved
		111b	Reserved
22:16	Reserved	Reserved.	
15:0	VendorID	This is as defined in the USB PD specification. The value written for VID here is used to populate the VID in other registers as well: TX_MIDB_SOP, TX_SCEDB, and TX_SKEDB.	
Byte 1: Header			
6	Reserved		
5:3	Reserved		
2:0	numValidVDOs	Number of valid VDO's in this register. This field causes special behavior: • When 0, the PD Controller will NAK USB PD Discover Identity message. • When 1, the PD Controller will respond with BUSY message. • When 2, the PD Controller will respond with Not_Supported (PD3) or no response (PD2). • When 3, 4, 5, or 6 the PD Controller will respond with an ACK message. • The value 7 is reserved. (Max of 6)	

Depending upon the product type the three VDO's in this register starting at byte 14 (ProductTypeVdo0SOP, ProductTypeVdo1SOP, ProductTypeVdo2SOP) have different meanings as defined by the USB PD specification. See the USB PD specification for definition of DFP VDO, UFP VDO1, and UFP VDO2.

Table 3-64. Description of product type VDO's.

DFP Product Type	UFP Product Type	numValidVDOs	ProductTypeVdo0 SOP	ProductTypeVdo1 SOP	ProductTypeVdo2 SOP
PDUSB Hub, PDUSB Host, Power Brick	Undefined, PSD	4	DFP VDO	N/A	N/A
	PDUSB Hub, PDUSB Peripheral	6	UFP VDO1	UFP VDO2	DFP VDO
Undefined	Undefined, PSD	3	N/A	N/A	N/A
	PDUSB Hub, PDUSB Peripheral	5	UFP VDO1	UFP VDO2	N/A

3.28 0x48 RX_IDENTITY_SOP Register

Table 3-65. 0x48 RX_IDENTITY_SOP Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x48	RX_IDENTITY_SOP	RO	25	yes	Cleared on connect.

Table 3-66. 0x48 RX_IDENTITY_SOP Register Bit Field Definitions

Bits	Name	Description	
Bytes 22-25: VDO #6 (treated as a 32-bit little endian value)			
31:0	rxIdSopVdo6	6th VDO. The sixth Data Object for Discover Identity response is context-specific.	
Bytes 18-21: VDO #5 (treated as a 32-bit little endian value)			
31:0	rxIdSopVdo5	5th VDO. The fifth Data Object for Discover Identity response is context-specific.	
Bytes 14-17: VDO #4 (treated as a 32-bit little endian value)			
31:0	rxIdSopVdo4	4th VDO. The fourth Data Object for Discover Identity response is context-specific as defined in USB PD.	
Bytes 10-13: VDO #3 (treated as a 32-bit little endian value)			
31:0	rxIdSopVdo3	Product VDO. The third Data Object for Discover Identity response.	
Bytes 6-9: VDO #2 (treated as a 32-bit little endian value)			
31:0	rxIdSopVdo2	Cert Stat VDO. The second Data Object for Discover Identity response.	
Bytes 2-5: VDO #1 (treated as a 32-bit little endian value)			
31:0	rxIdSopVdo1	ID Header VDO. The first Data Object in Discover Identity response.	
Byte 1: Header			
7:6	ResponseType	Type of response received.	
		00b	SOP Discover Identity REQ not sent or pending.
		01b	Responder ACK received.
		10b	Responder NAK received or response timeout.
		11b	Responder BUSY received. The PD Controller will retry.
5:3	Reserved		
2:0	numValidVDOs	Number of valid VDO's in this register. (Max of 6)	

3.29 0x49 RX_IDENTITY_SOPp Register

Table 3-67. 0x49 RX_IDENTITY_SOPp Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x49	RX_IDENTITY_SOPp	RO	25	yes	Cleared on connect.

Table 3-68. 0x49 RX_IDENTITY_SOPp Register Bit Field Definitions

Bits	Name	Description	
Bytes 22-25: VDO #6 (treated as a 32-bit little endian value)			
31:0	rxIdSoppVdo6	6th VDO. The sixth Data Object for Discover Identity response is context-specific.	
Bytes 18-21: VDO #5 (treated as a 32-bit little endian value)			
31:0	rxIdSoppVdo5	5th VDO. The fifth Data Object for Discover Identity response is context-specific.	
Bytes 14-17: VDO #4 (treated as a 32-bit little endian value)			
31:0	rxIdSoppVdo4	4th VDO. The fourth Data Object for Discover Identity response is context-specific as defined in USB PD.	
Bytes 10-13: VDO #3 (treated as a 32-bit little endian value)			
31:0	rxIdSoppVdo3	Product VDO. The third Data Object for Discover Identity response.	
Bytes 6-9: VDO #2 (treated as a 32-bit little endian value)			
31:0	rxIdSoppVdo2	Cert Stat VDO. The second Data Object for Discover Identity response.	
Bytes 2-5: VDO #1 (treated as a 32-bit little endian value)			
31:0	rxIdSoppVdo1	ID Header VDO. The first Data Object in Discover Identity response.	
Byte 1: Header			
7:6	ResponseType	Type of response received.	
		00b	SOP' Discover Identity REQ not sent or pending.
		01b	Responder ACK received.
		10b	Responder NAK received or response timeout.
		11b	Responder BUSY received. The PD Controller will retry.
5:3	Reserved		
2:0	numValidVDOs	Number of valid VDO's in this register. (Max of 6)	

3.30 0x4A USER_VID_CONFIG Register

This register is sampled each time the PD controller needs the information. For example, each time the PD controller implements the discovery process it checks the contents of this register and carries out the discovery process accordingly.

Table 3-69. 0x4A USER_VID_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x4A	USER_VID_CONFIG	RW	63	yes	Initialized by Application Configuration

Table 3-70. 0x4A USER_VID_CONFIG Register Bit Field Definitions

Bits	Name	Description
Byte 63: Number of VDO's for automatic transmission		
7:0	UserModeAutoSendVdoCount	Number of VDOs in UVDM. If auto send unstructured VDM enabled, number of VDOs to send. (Max of 6)
Bytes 61-62: Additional data to use for automatic transmission		
15:14	Reserved	
13:0	UserModeAutoSendVendorUseData	Additional data for UVDM. If auto send unstructured VDM enabled, up to an additional 14 bits may be sent as part of the Unstructured VDM header.
Bytes 37-60: Data to use for automatic transmission		
191:0	UserModeAutoSendVdoData	Data for UVDM. If auto send unstructured VDM enabled, up to 24 bytes may be sent.
Bytes 33-36: Mode 4 Name		
31:0	UserAltModeMode4Value	User VID mode 4 name.
Bytes 29-32: Mode 3 Name		
31:0	UserAltModeMode3Value	User VID mode 3 name.
Bytes 25-28: Mode 2 Name		
31:0	UserAltModeMode2Value	User VID mode 2 name.
Bytes 21-24: Mode 1 Name		
31:0	UserAltModeMode1Value	User VID mode 1 name.
Bytes 17-20: AppConfig Data for each Mode		
31:25	Reserved	
24	UserMode4LoadAppConfigData	Load App Config upon entry to User VID mode 4. Assert this bit to load application data upon entry to User VID mode 4.
23:17	Reserved	
16	UserMode3LoadAppConfigData	Load App Config upon entry to User VID mode 3. Assert this bit to load application data upon entry to User VID mode 3.
15:9	Reserved	
8	UserMode2LoadAppConfigData	Load App Config upon entry to User VID mode 2. Assert this bit to load application data upon entry to User VID mode 2.
7:1	Reserved	
0	UserMode1LoadAppConfigData	Load App Config upon entry to User VID mode 1. Assert this bit to load application data upon entry to User VID mode 1.
Bytes 13-16: Automatic Sending of Unstructured VDM Configuration		
31:25	Reserved	
24	UserMode4AutoSendUnstrcVdm	Enable auto-sending of UVDM for User VID mode 4.
23:17	Reserved	
16	UserMode3AutoSendUnstrcVdm	Enable auto-sending of UVDM for User VID mode 3.
15:9	Reserved	
8	UserMode2AutoSendUnstrcVdm	Enable auto-sending of UVDM for User VID mode 2.
7:1	Reserved	
0	UserMode1AutoSendUnstrcVdm	Enable auto-sending of UVDM for User VID mode 1.

Table 3-70. 0x4A USER_VID_CONFIG Register Bit Field Definitions (continued)

Bits	Name	Description
Bytes 9-12: Automatic Mode Entry Configuration		
31:25	Reserved	
24	UserMode4AutoModeEntryAllowed	Enable auto-entry for User VID mode 4.
23:17	Reserved	
16	UserMode3AutoModeEntryAllowed	Enable auto-entry for User VID mode 3.
15:9	Reserved	
8	UserMode2AutoModeEntryAllowed	Enable auto-entry for User VID mode 2.
7:1	Reserved	
0	UserMode1AutoModeEntryAllowed	Enable auto-entry for User VID mode 1.
Bytes 5-8: Mode Enable Selection		
31:25	Reserved	
24	UserMode4Enabled	Assert this bit to enable User VID mode 4.
23:17	Reserved	
16	UserMode3Enabled	Assert this bit to enable User VID mode 3.
15:9	Reserved	
8	UserMode2Enabled	Assert this bit to enable User VID mode 2.
7:1	Reserved	
0	UserMode1Enabled	Assert this bit to enable User VID mode 1.
Bytes 3-4: SVID value		
15:0	User_AltMode_SVID_Value	User VID
Bytes 1-2: Enabling Configuration		
15:8	Reserved	
7:2	Reserved	Write as zero.
0	UserVidEnabled	Assert this bit to enable User VID.

3.31 0x4B MIPI_VID_CONFIG Register

This register is sampled each time the PD controller needs the information. For example, each time the PD controller implements the discovery process it checks the contents of this register and carries out the discovery process accordingly.

Table 3-71. 0x4B MIPI_VID_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x4B	MIPI_VID_CONFIG	RW	4	yes	Cleared on connect.

Table 3-72. 0x4B MIPI_VID_CONFIG Register Bit Field Definitions

Bits	Name	Description
31:16	overlaySupport	
15:10	Reserved	
9	MipiDebugModeAutoEntryAllowed	Enable automatic entry for MIPI mode. When this bit is asserted, the PD controller will automatically enter the MIPI mode whenever possible.
8:2	Reserved	
1	MipiDebugModeEnabled	Enable Debug mode.
0	MipiVidEnabled	Enable MIPI mode.

3.32 0x4E RX_ATTENTION_VDM Register

The PD controller does not queue received Attention messages. This register contains the most recently received message.

Only Structured VDM messages with Command Type = Initiator (00b) and Command = Attention (6) get stored in this buffer. See register 0x4F for all other received VDMs.

Table 3-73. 0x4E RX_ATTENTION_VDM Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x4E	RX_ATTENTION_VDM	RO	9	yes	Cleared on disconnect or Hard Reset

Table 3-74. 0x4E RX_ATTENTION_VDM Register Bit Field Definitions

Bits	Name	Description
Bytes 6-9: DO #1 (treated as a 32-bit little endian value)		
31:0	rxVDMs2	Second data object. The vendor defined object (VDO) received with the Attention message (if any).
Bytes 2-5: DO #1 (treated as a 32-bit little endian value)		
31:0	rxVDMs1	Structured VDM header. This contains the first Data Object of most recently received Attention SVDM.
Byte 1: Rx Attention Status		
7:5	seqNum	Sequence number. This field increments by one every time this register is updated, rolls over upon reflow
4:3	Reserved	
2:0	numValidDOs	Number of valid data objects in this register.

3.33 0x4F RX_OTHER_VDM Register

The PD controller does not queue received SVDM messages. This register contains the most recently received message.

Structured VDM "Attention" Initiator messages are only stored in register 0x4E, not this register.

Table 3-75. 0x4F RX_OTHER_VDM Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x4F	RX_OTHER_VDM	RO	29	yes	Cleared on disconnect or Hard Reset

Table 3-76. 0x4F RX_OTHER_VDM Register Bit Field Definitions

Bits	Name	Description
Bytes 26-29: DO #7 (treated as a 32-bit little endian value)		
31:0	rxVDMs7	Seventh Data Object of most recently received VDM.
Bytes 22-25: DO #6 (treated as a 32-bit little endian value)		
31:0	rxVDMs6	Sixth Data Object of most recently received VDM.
Bytes 18-21: DO #5 (treated as a 32-bit little endian value)		
31:0	rxVDMs5	Fifth Data Object of most recently received VDM.
Bytes 14-17: DO #4 (treated as a 32-bit little endian value)		
31:0	rxVDMs4	Fourth Data Object of most recently received VDM.
Bytes 10-13: DO #3 (treated as a 32-bit little endian value)		
31:0	rxVDMs3	Third Data Object of most recently received VDM.
Bytes 6-9: DO #2 (treated as a 32-bit little endian value)		
31:0	rxVDMs2	Second Data Object of most recently received VDM.
Bytes 2-5: DO #1 (treated as a 32-bit little endian value)		
31:0	rxVDMs1	First Data Object of most recently received VDM.
Byte 1: Rx VDM Status		
7:5	seqNum	
4:3	sopType	
2:0	numValidVDOs	

3.34 0x50 DATA_CONTROL Register

This register provides *shortcuts* that set other bits in other registers, for convenience of the Thunderbolt Controller or other host. It also holds data that the TBT controller wants to write to the Debug Mode Register of the Retimer through the PD controller. The TBT Controller initiates the PD controller's write to the Debug Mode Register of the Retimer using the DATA_CONTROL Register as well.

NOTE: In systems with an EC a second host (eg. TBT controller or SoC) that requires access to this register, the EC should not write to this register since it may cause a conflict with the other host.

Table 3-77. 0x50 DATA_CONTROL Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x50	DATA_CONTROL	RW	6	yes	0 (Managed by Thunderbolt Controller)

Table 3-78. 0x50 DATA_CONTROL Register Bit Field Definitions

Bits	Name	Description
Bytes 3-6: Retimer Debug Mode Register		
31	TraceMode	
30	Enable	
29:28	DebugModeType	
26:24	AdditionalSkew	
23:22	LaneCount	
21:20	Freq	
19:18	EmphasisDB	
17:16	AmpDB	
15	DebugRxLocked	
7:6	TPSType	
5:4	RequestedWindowSize	
3:2	EmphasisIcl	
1:0	AmpIcl	
Bytes 1-2: Data Control (treated as 16-bit little endian value)		
15	HPD_Level[1][2]	HPD level from DP Sink connection to Titan Ridge to PD Controller. Used only for TBTdevices.Not used for TBT hosts. 0b HPD low. 1b HPD high.
14	HPD_IRQ_Sticky[1][2]	HPD IRQ from DP Sink connection to Titan Ridge to PD Controller. Cleared when DATA_STATUS.HPD_IRQ_Sticky. Used only for TBT devices. Not used for TBT hosts. 0b No HPD_IRQ_ACK 1b HPD_IRQ_ACK
13	IRQ_ACK	Writing this bit as 1 will clear DATA_STATUS.HpdIrqSticky after a delay of DELAY_CONFIG.HpdDelay.
12	WriteToRetimer	If asserted the PD controller will write bytes 3-6 to to Retimer on I2C3m.
9:6	Reserved	
7	PowerReset	If asserted the PD controller will initiate Error Recovery on the connector.
6	DataReset	If asserted the PD controller will initiate Data Reset process.
5	UsbHostConnected[1][2]	
4	DpHostConnected[1][2]	
5:3	Reserved	Reserved (Host may write 0 or 1, no action to be taken).

[1] This feature is not supported by TPS65994AD_F509.04.02.

[2] This feature is not supported by TPS65993AD_F509.04.02.

Table 3-78. 0x50 DATA_CONTROL Register Bit Field Definitions (continued)

Bits	Name	Description
2	InterruptAck	When set, causes INT_MASK2 value to be written to INT_CLEAR2 (clearing all interrupt events).
1	SoftReset	When set, causes a soft-reset of PD Controller. Equivalent to Gaid 4CC.
0	HostConnected	Assert this bit when a TBT host is connected. The TBT controller may also assert this bit to 1 then 0 to force a port disconnect/reconnect.

3.35 0x51 DP_SID_CONFIG Register

This register is sampled each time the PD controller needs the information. For example, each time the PD controller implements the discovery process it checks the contents of this register and carries out the discovery process accordingly.

Table 3-79. 0x51 DP_SID_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x51	DP_SID_CONFIG	RW	6	yes	Initialized by Application Configuration

Table 3-80. 0x51 DP_SID_CONFIG Register Bit Field Definitions

Bits	Name	Description	
Bytes 5-6: Additional Configuration			
15:9	Reserved		
8	DPModeAutoEntryAllowed	Assert this bit to enable auto-entry.	
7:5	Reserved[1][2]		
4:3	DfpdUfpdConnected[1][2]	This field indicates the status of the connection.	
		00b	Neither UFP_D nor DFP_D is connected. May also be that the adapter is disabled.
		01b	DFP_D is connected.
		10b	UFP_D is connected
		11b	Both UFP_D and DFP_D is connected.
2:1	Reserved		
0	multifunctionPreferred	Assert this bit if multi-function is preferred.	
Bytes 1-4: DisplayPort Configuration			
31:24	UfpD_PinAssignment[1][2]	UFP_D Pin Assignments Supported. Each bit corresponds to an allowed pin assignment. Multiple pin assignments may be allowed.	
		00000000b	UFP pin assignments are not supported.
		xxxxxxx1b	Pin assignment A is supported
		xxxxxx1xb	Pin assignment B is supported
		xxxxx1xxb	Pin assignment C is supported
		xxxx1xxxb	Pin assignment D is supported
		xxx1xxxxb	Pin assignment E is supported
		xx1xxxxxb	Reserved
		x1xxxxxxb	Reserved
		1xxxxxxxb	Reserved
23:16	DfpD_PinAssignment	DFP_D Pin Assignments Supported. Each bit corresponds to an allowed pin assignment. Multiple pin assignments may be allowed.	
		00000000b	DFP pin assignments are not supported.
		xxxxxxx1b	Pin assignment A is supported
		xxxxxx1xb	Pin assignment B is supported
		xxxxx1xxb	Pin assignment C is supported
		xxxx1xxxb	Pin assignment D is supported
		xxx1xxxxb	Pin assignment E is supported
		xx1xxxxxb	Pin assignment F is supported
		x1xxxxxxb	Reserved
		1xxxxxxxb	Reserved

[1] This feature is not supported by TPS65994AD_F509.04.02.

[2] This feature is not supported by TPS65993AD_F509.04.02.

Table 3-80. 0x51 DP_SID_CONFIG Register Bit Field Definitions (continued)

Bits	Name	Description	
15	usb2SignallingNotUsed	USB r2.0 signaling requirement on A6 - A7 or B6 - B7 (D+/D-) while in DP configuration	
		0b	May be required.
		1b	Not required.
14	Reserved		
13:10	DpTransportSignaling	Signaling for transport of DisplayPort protocol.	
		0h	Supports USB signaling rate and electrical specification
		1h	Supports DP signaling rates and electrical specification
		2h-Fh	Reserved
9:8	DPPortCapability	Display port capabilities	
		00b	Reserved
		01b	UFP_D-capable (including Branch device)
		10b	DFP_D-capable (including Branch device)
		11b	Reserved
7:2	Reserved		
1	DPMode	Assert this bit to enable DisplayPort Alternate mode.	
0	enableDPSID	Assert this bit to enable DisplayPort SVID.	

3.36 0x52 INTEL_VID_CONFIG Register

This register is sampled each time the PD controller needs the information. For example, each time the PD controller implements the discovery process it checks the contents of this register and carries out the discovery process accordingly.

Table 3-81. 0x52 INTEL_VID_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x52	INTEL_VID_CONFIG	RW	7	yes	Initialized by Application Configuration

Table 3-82. 0x52 INTEL_VID_CONFIG Register Bit Field Definitions

Bits	Name	Description
Bytes 5-7: Additional Configuration		
23:18	Reserved	
17	thunderBoltAutoEntryAllowed	Assert this bit to enable TBT auto-entry.
16	Reserved	
15:0	Reserved	
Bytes 3-4: TBT Mode SOP Data (treated as 16-bit little endian value). Upper 16 bits of data to be sent on Intel VID SVDM Discover Modes (SOP) response (UFP) or used to drive TBT AM policy (DFP).		
15:11	Reserved	
10	VproSupported	
9:1	Reserved	
0	LegacyTbtAdapter	
Byte 2: Intel Mode Configuration		
7	RetimerComplianceSupport	If this bit is set to 1b, then asserting the Retimer_SOC_OVR_Force_Power GPIO causes the PD controller to place an attached Intel Retimer into compliance mode. De-asserting the Retimer_SOC_OVR_Force_Power GPIO places the retimer into normal operation.
6	DataStatusHPDEvents	This bit controls how HPD events are configured. 0b HPD over GPIO. HPD events are reported using the GPIO configured as HPD. 1b Virtual HPD. HPD events are reported in DATA_STATUS.HpdLevel and HpdIrqSticky
5	TBTRetimerPresent	Enable first retimer on each port. Assert this bit when there is a TBT retimer on this port.
4	DualTBTRetimerPresent	Enable second retimer on each port. Assert this bit when there is a second TBT retimer on this port.
3	Reserved	
2	ANMinPowerRequired	Power required for TBT mode entry. 0b Power does not matter. TBT Mode can be configured regardless of power contract. DATA_STATUS.CapabilityMismatch will always be 0. 1b TBT not fully connected unless power is available. PD Controller will enter TBT Mode regardless of power contract, but will not set Data Status to indicate TBT Connected until ANMinimumPower limit has been met. Note that once sufficient power is available, the Billboard will not be re-enabled even if a later contract does not provide sufficient power, however the DATA_STATUS.CapabilityMismatch always reflects active contract when PD Controller is a Sink.

Table 3-82. 0x52 INTEL_VID_CONFIG Register Bit Field Definitions (continued)

Bits	Name	Description	
1	TbtEMarkerOverride	Configuration for non-responsive Cable Plug.	
		0b	Assume non-TBT cable. If the PD Controller does not receive a response on SOP' it will assume a Cable VDO b2:0 value of 000b, which prevents Thunderbolt Mode entry.
		1b	Assume TBT cable. If the PD Controller does not receive a response on SOP' it will assume a Cable VDO b2:0 value of 001b, which allows Thunderbolt Mode entry (will treat cable as 10Gb/s passive).
Byte 1: Intel Mode Enables			
7:2	Reserved		
1	thunderBoltModeEnabled	Assert this bit to enable TBT mode. When this bit is asserted The PD Controller as UFP will advertise Thunderbolt Mode. The PD Controller as DFP will negotiate Thunderbolt Mode.	
0	IntelVidEnabled	Assert this bit to enable Intel VID.	

3.37 0x57 USER_VID_STATUS Register

Table 3-83. 0x57 USER_VID_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x57	USER_VID_STATUS	RO	2	yes	Cleared on disconnect or Hard Reset.

Table 3-84. 0x57 USER_VID_STATUS Register Bit Field Definitions

Bits	Name	Description
Byte 2: Mode Status		
7:5	Reserved	
4	Mode4Status	Asserted when Mode4 has been entered.
3	Mode3Status	Asserted when Mode3 has been entered.
2	Mode2Status	Asserted when Mode2 has been entered.
1	Mode1Status	Asserted when Mode1 has been entered.
0	Reserved	
Byte 1: VID status		
7:5	Reserved	
4:2	User_VIDErrorCode	Error code
1	User_VIDActive	Asserted when a User VID is active.
0	User_VIDDetected	Asserted when a User VID has been detected.

3.38 0x58 DP_SID_STATUS Register

Table 3-85. 0x58 DP_SID_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x58	DP_SID_STATUS	RO	37	yes	Cleared on disconnect or Hard Reset.

Table 3-86. 0x58 DP_SID_STATUS Register Bit Field Definitions

Bits	Name	Description
Bytes 34-37: DP Mode Data RX SOP' (treated as 32-bit little endian value)		
31:0	DPModeDataSOPp	Discover Mode ACK (SOP'). This field contains the contents of DP Discover Mode response from Cable Plug
Bytes 30-33: DP Configure RX SOP' (treated as 32-bit little endian value)		
31:0	DPConfigFromPlug	Received DP config message (SOP'). This field contains the contents of DP Configure message received from Cable Plug
Bytes 26-29: DP Configure TX SOP' (treated as 32-bit little endian value)		
31:0	DPConfigToPlug	DP Config message (SOP'). This field contains the contents of DP Configure message sent to Cable Plug
Bytes 22-25: DP Status RX SOP' (treated as 32-bit little endian value)		
31:0	DPStatusAckFromPlug	DP Status acknowledgement (SOP'). This field contains the most recently received DP Status Acknowledgment from Cable Plug
Bytes 18-21: DP Status TX SOP' (treated as 32-bit little endian value)		
31:0	DPStatusToPlug	DP Status (SOP'). This field contains the current Outgoing DP Status message contents to Cable Plug
Bytes 14-17: DP Mode Data TX/RX (treated as 32-bit little endian value)		
31:0	DPModeData	DP Discover Mode response. This field contains the contents of the DP Discover Mode response when received (DFP_U). Or sent.(UFP_U).
Bytes 10-13: DP Configure TX/RX (treated as 32-bit little endian value)		
31:0	DPConfigureMessage	Contents of DP Config message when sent (DFP_U).Or received (UFP_U).
Bytes 6-9: DP Status RX (treated as 32-bit little endian value)		
31:0	DPStatusRx	Most recently received DP Status message contents.
Bytes 2-5: DP Status TX (treated as 32-bit little endian value)		
31:0	DPStatusTx	Current Outgoing DP Status message contents
Byte 1: DP Mode Status		
7:5	Reserved	
4:2	Reserved	
1	DPModeActive	DP mode entered. DFP_U: This bit is asserted when PD Controller has entered DisplayPort Mode with attached UFP_U. UFP_U: Attached DFP_U has entered DisplayPort Mode.
1	DPModeActive	DP mode entered. This bit is asserted when PD Controller has entered DisplayPort Mode with attached UFP_U.
0	DPSidDetected	Port Partner is Display Port capable. This bit is asserted when UFP_U returns DP SID in Discover SVIDs response or responded with ACK to DP SID SVDM Commands.

3.39 0x59 INTEL_VID_STATUS Register

Table 3-87. 0x59 INTEL_VID_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x59	INTEL_VID_STATUS	RO	11	yes	Cleared on disconnect or Hard Reset.

Table 3-88. 0x59 INTEL_VID_STATUS Register Bit Field Definitions

Bits	Name	Description	
Bytes 10-11: TBT Discover Modes Response SOP' (treated as 16-bit little endian value)			
15:0	TBTModeDataRxOnSOPp	Data for Discover Modes (SOP'). This field contains the upper 16 bits of SOP' Discover Modes Cable response for TBT Mode. Lower 16 bits of the response are always 0x0001	
Bytes 8-9: TBT Discover Modes Response SOP (treated as 16-bit little endian value)			
15:0	TBTModeDataRxOnSOP	Data for Discover Modes response. This field contains the upper 16 bits of SOP Discover Modes response for TBT Mode when received (DFP) or sent (UFP). Lower 16 bits of the response are always 0x0001. NOTE: In the UFP role, this register simply copies the contents of the Intel VID Configuration register bits 23:8 at the time the Discover Modes response is generated.	
Bytes 6-7: TBT Enter Mode TX/RX (treated as 16-bit little endian value)			
15:0	TBTEnterModeData	Data for TBT Enter mode message. This field contains the upper 16 bits of second VDO to Thunderbolt Enter Mode command when sent (DFP) or received (UFP).	
Bytes 2-5: TBT Attention TX/RX (treated as 32-bit little endian value)			
31:0	TBTAttnData	Attention message contents. This field contains the contents of the Attention VDO in Thunderbolt Mode when sent (UFP) or received (DFP).	
Byte 1: Intel Mode Status			
7:3	Reserved		
2	ForcedTbtMode	Retimer in TBT state and ready for FW update. A value of 1 indicates that the PD controller has placed the retimer in the TBT state and it is ready for FW update.	
1	TBTModeActive	DFP	TBT mode is entered as DFP. This bit is asserted when PD Controller has entered Thunderbolt Mode and the Port Partner is the UFP.
		UFP	TBT mode is entered as UFP. This bit is asserted when PD controller has entered Thunderbolt Mode and the Port Partner is the DFP.
0	IntelVidDetected	DFP	Sent Intel SVID to the Port Partner. The PD controller in the UFP data-role has returned Intel VID in Discover SVIDs response or responded with ACK to Intel VID SVDM Commands.
		UFP	The PD controller has discovered modes. The PD controller in the DFP data-role has issued Intel VID SVDM (Discover Modes, Enter Mode, etc).

3.40 0x5C GPIO_CONFIG Register

The figure below shows the interface for the GPIO hardware. The register fields GPIO_AI_EN, GPIO_PD_EN, GPIO_PU_EN, GPIO_OE, and GPIO_OD_EN listed in the table below are passed along to the GPIO hardware, independent of the GPIO Event that is configured. The PD controller will then set the signal GPIO_O to high or low to implement the selected output GPIO event. So for example, each output GPIO Event is configurable as push-pull or open-drain using the GPIO_OD_EN bit. For input GPIO events the PD controller will monitor GPIO_I from the figure.

NOTE: A given GPIO Event can only be assigned to one GPIO pin.

The ProcHot_N_Event GPIO Event can be assigned to any GPIO, but for some PD controllers only a specific GPIO will have the fast reaction time feature (see device datasheet).

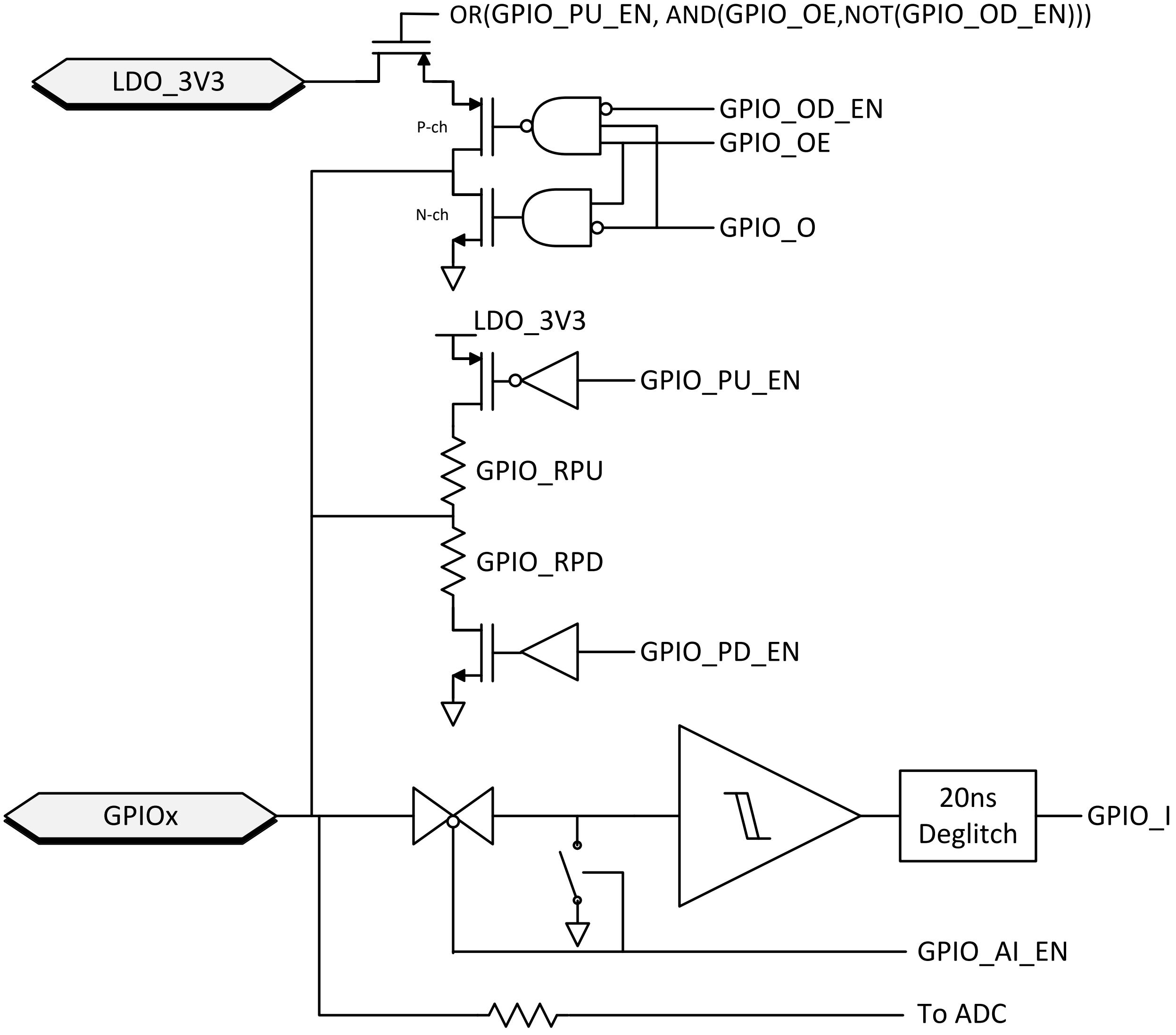


Figure 3-2. Interface to GPIO hardware.

Table 3-89. 0x5C GPIO_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x5C	GPIO_CONFIG	RO	49	no	Initialized by Application Configuration

Table 3-90. 0x5C GPIO_CONFIG Register Bit Field Definitions

Bits	Name	Description
Bytes 37-49: GPIO Event configurations		
103	Reserved	
102:96	GPIO12Event	Event table mapping for GPIO12. See Table 3-94.
95	Reserved	
94:88	GPIO11Event	Event table mapping for GPIO11. See Table 3-94.
87	Reserved	
86:80	GPIO10Event	Event table mapping for GPIO10. See Table 3-94.
79	Reserved	
78:72	GPIO9Event[1]	Event table mapping for GPIO9. See Table 3-94.
71	Reserved	
70:64	GPIO8Event[1]	Event table mapping for GPIO8. See Table 3-94.
63	Reserved	
62:56	GPIO7Event	Event table mapping for GPIO7. See Table 3-94.
55	Reserved	
54:48	GPIO6Event	Event table mapping for GPIO6. See Table 3-94.
47	Reserved	
46:40	GPIO5Event	Event table mapping for GPIO5. See Table 3-94.
39	Reserved	
38:32	GPIO4Event	Event table mapping for GPIO4. See Table 3-94.
31	Reserved	
30:24	GPIO3Event	Event table mapping for GPIO3. See Table 3-94.
23	Reserved	
22:16	GPIO2Event	Event table mapping for GPIO2. See Table 3-94.
15	Reserved	
14:8	GPIO1Event	Event table mapping for GPIO1. See Table 3-94.
7	Reserved	
6:0	GPIO0Event	Event table mapping for GPIO0. See Table 3-94.
Bytes 33-36: GPIO Event polarity configuration		
31:13	Reserved	
12:0	GPIOEventPol	Controls polarity of a selected output event for each GPIO. Assert the bit for a given GPIO to invert the polarity of the event mapped to it. This field has no impact for input GPIO Events.
Bytes 29-32: GPIO Analog input configuration		
31:6	Reserved	
5	GPIO_AI_EN_GPIO5	Assert when GPIO5 is used as an analog input. This must also be asserted when PORT_CONTROL.ChargerDetectEnable or ChargerAdvertiseEnable is non-zero.
4	GPIO_AI_EN_GPIO4	Assert when GPIO4 is used as an analog input. This must also be asserted when PORT_CONTROL.ChargerDetectEnable or ChargerAdvertiseEnable is non-zero.
3	Reserved	
2	GPIO_AI_EN_GPIO2[2][3]	Assert when GPIO4 is used as an analog input.
1	Reserved	

[1] This feature is not supported by TPS65992xAD_F509.05.02.
[2] This feature is not supported by TPS65994AD_F509.04.02.
[3] This feature is not supported by TPS65993AD_F509.04.02.

Table 3-90. 0x5C GPIO_CONFIG Register Bit Field Definitions (continued)

Bits	Name	Description
0	GPIO_AI_EN_GPIO0[2][3]	Assert when GPIO4 is used as an analog input.
Bytes 25-28: GPIO Pull-up enable		
31:13	Reserved	
12:0	GPIO_PU_EN	Controls weak pull-up setting for each configurable GPIO (1=Enabled, 0=Disabled).
Bytes 21-24: GPIO pull-down enable		
31:13	Reserved	
12:0	GPIO_PD_EN	Controls weak pull-down setting for each configurable GPIO (1=Enabled, 0=Disabled).
Bytes 17-20: GPIO Secondary mux functionality configuration		
31:12	Reserved	
11:0	GPIOMuxSel	Selects between GPIO (0) and Secondary Function (1) for each GPIO. See the subsection below for more details.
Bytes 13-16: GPIO Open-Drain Enable		
31:13	Reserved	
12:0	GPIO_OD_EN	Controls push-pull (0) vs. open-drain (1) setting for each configurable GPIO.
Bytes 9-12: Output Controls		
31:13	Reserved	
12:0	GPIOData	Controls default output level for each GPIO enabled as output (0=Drive Low, 1=Drive High)
Bytes 5-8: Interrupt Enable		
31:13	Reserved	
12:0	GPIOIntEn	Controls interrupt enable for each GPIO (1=Interrupt Enabled, 0=Interrupt Disabled). Note that all GPIO pins may not be configured as inputs (see the data-sheet).
Bytes 1-4: Output Enable		
31:13	Reserved	
12:0	GPIO_OE	Controls output enable for each GPIO (1=Output Enabled, 0=Hi-Z). Note that all GPIO may not be configurable as an output (see data-sheet).

3.40.1 GPIO Multiplexor

Table 3-93 defines the GPIO Secondary Functions supported used in bytes 17-20 of the I/O Configuration Register.

Table 3-91. GPIO Secondary Functions (TPS65994AD)

Bit #	Secondary Function Name	I/O	Description
11	I2C2s_IRQ	Output	Alert signal for the I2C2s bus.
10	I2C1s_IRQ	Output	Alert signal for the I2C_EC bus.
1	HPD_TX1	Output	HPD_Tx for PortA
0	HPD_TX2	Output	HPD_Tx for PortB

Table 3-92. GPIO Secondary Functions (TPS65993AD)

Bit #	Secondary Function Name	I/O	Description
11	I2C2s_IRQ	Output	Alert signal for the I2C2s bus.
10	I2C1s_IRQ	Output	Alert signal for the I2C_EC bus.
1	HPD_TX1	Output	HPD_Tx for PortA

Table 3-93. GPIO Secondary Functions (TPS65992xAD)

Bit #	Secondary Function Name	I/O	Description
11	I2C2s_IRQ	Output	Alert signal for the I2C2s bus.
10	I2C1s_IRQ	Output	Alert signal for the I2C_EC bus.
5	DM	I/O	Connect to the D- pin for BC1.2 functionality.
4	DP	I/O	Connect to the D+ pin for BC1.2 functionality.
3	HPD_RX1	Input	HPD_Rx for PortA
1	HPD_TX1	Output	HPD_Tx for PortA

3.40.2 GPIO Events

Table 3-94. GPIO Events

Event #	Event Name	I/O	Description
76	PdNegotiationInProcess	Output	When in source mode, this GPIO is asserted after a Request message is received, before sending the Accept message. The GPIO is de-asserted after the PS_RDY message is sent. When in sink mode, this GPIO is asserted right before sending a Request message, and de-asserted after a PS_RDY message is received. In either mode, the GPIO is de-asserted when a detach occurs.
75	AttachedAsSink	Output	When the PD controller has a port that is connected to a Source, this GPIO will be asserted. The GPIO is de-asserted upon disconnect, hard reset, during power-role swap and during fast-role swap only if none of the ports in the PD controller are connected to a source.
74	EnableSource_Port2[(1)]	Output	It has the same behavior as EnableSource_Port1, but it applies to Port 2.
73	EnableSource_Port1	Output	When the PD controller sends an Accept message to start sourcing VBUS on Port1 under a high-power contract this GPIO is asserted high. It will remain high as long as the high-power contract is active. If the contract transitions from a high-power contract to a low-power contract, this GPIO will have a high-to-low transition after the PS_Rdy message is sent. A high-power contract is one for a PDO index greater than 1, or a current greater than GLOBAL_SYSTEM_CONFIG.MinimumCurrentAdvertisement.
72	Reserved		
71	Reserved		
70	Reserved		
69	MRESET	Input	Upon a rising edge on this GPIO the PD controller will drive a rising edge on the RESETZ GPIO after a delay. Upon a falling edge on this GPIO the PD controller will drive a falling edge on the RESETZ GPIO after a delay.
68	RESETZ	Output	This works in conjunction with MRESET.
67	Fault_Condition_Active_Low_Global[(1)]	Output	Asserts low on an overcurrent event on Port1 or Port2.
66	Load_Switch_Drive_Port2[(2)][(1)]	Output	When the PD controller enables the PP_EXT2 sinking path, it will pull the selected GPIO low to enable a load-switch. When the PD controller disables the PP_EXT2 sinking path, it will drive the selected GPIO high.
65	Load_Switch_Drive_Port1	Output	When the PD controller enables the PP_EXT1 sinking path, it will pull the selected GPIO low to enable a load-switch. When the PD controller disables the PP_EXT1 sinking path, it will drive the selected GPIO high.

[(1)] This feature is not supported by TPS65992xAD_F509.05.02.
[(2)] This feature is not supported by TPS65993AD_F509.04.02.

Table 3-94. GPIO Events (continued)

Event #	Event Name	I/O	Description
64	Reserved		
63	Reserved		
62	Dp_Dm_Mux_Enable_Event_Port2 [2] [1]	Output	This GPIO should be used to enable/disable a USB 2.0 D+/D- mux. The GPIO is driven high upon connection, and low upon disconnect on Port2.
61	Dp_Dm_Mux_Enable_Event_Port1	Output	This GPIO should be used to enable/disable a USB 2.0 D+/D- mux. The GPIO is driven high upon connection, and low upon disconnect on Port1.
60	Reserved		
59	Reserved		
58	Reserved		
57	Reserved		
56	I2C3_MASTER_IRQ_EVENT	Input	When this GPIO is pulled low, the PD controller copies the contents of register 0x5F, clears bits 3, 4, 13, 22, 24, and 30, then writes the resulting 4 bytes to register address 0x04 using the slave address specified in the SlaveAddrTbt2 field of the I2CMASTER_CONFIG register (0x64). The PD controller always does this for both ports.
55	Prs_Ext_Vbus_Discharge_Event_Port2 [2] [1]	Output	This GPIO is pulled low after a PR_Swap to enable an external VBUS discharge circuit during a power-role swap on Port2.
54	Prs_Ext_Vbus_Discharge_Event_Port1	Output	This GPIO is pulled low after a PR_Swap to enable an external VBUS discharge circuit during a power-role swap on Port1.
53	VCONN_On_Event_Port2 [2] [1]	Output	This GPIO is asserted when PP_CABLE2 is enabled.
52	VCONN_On_Event_Port1	Output	This GPIO is asserted when PP_CABLE1 is enabled.
51	Debug_Accessory_Mode_Event_Port2 [2] [1]	Output	Output: This GPIO is asserted high when an Debug Accessory is attached on Port2.
50	Debug_Accessory_Mode_Event_Port1	Output	Output: This GPIO is asserted high when a Debug Accessory is attached on Port1.
49	Audio_Mode_Event_Port2 [2] [1]	Output	Output: This GPIO is asserted high when an Audio Accessory (Ra/Ra) is attached on Port2.
48	Audio_Mode_Event_Port1	Output	Output: This GPIO is asserted high when an Audio Accessory (Ra/Ra) is attached on Port1.
47	Prevent_High_Current_Contract_Event	Input	When this GPIO is high, then the PD controller will limit the amount of current it advertises to the value specified by the MinimumCurrentAdvertisement field in the Global System Configuration register (0x27) and it will only advertise one PDO. The PD controller responds to both falling and rising edges of this GPIO by increasing or decreasing its current advertisement. The adjusted current advertisement is reflected by automatically transmitting an updated Source Capabilities message for explicit contracts and by adjusting the Type-C Rp value for implicit contracts.
46	High_current_Contract_Active_Event [2] [1]	Output	When an explicit contract for more than the value specified by the MinimumCurrentAdvertisement field in the Global System Configuration register (0x27) is negotiated, the PD controller drives this GPIO high. When connected to a legacy device/sink this GPIO will not be driven high.
45	Prevent_DRSwap_To_UFP_Event	Input	When the GPIO is high, the PD controller will reject any DR_Swap messages from the Port Partner requesting to change the data-role from DFP to UFP.
44	UFP_Indicator_Event	Output	The GPIO is driven high when the data role of any port in the PD controller is UFP.

Table 3-94. GPIO Events (continued)

Event #	Event Name	I/O	Description
43	Barrel_Jack_Event	Input	When this GPIO is high, the PD controller interprets it to mean that a barrel-jack adaptor is connected and the system has Unconstrained power. A falling edge on this GPIO will automatically set PORT_CONTROL.UnconstrainedPower to 0 and TX_SCEDB.SourceInputs[0] to 0. A rising edge on this GPIO will automatically set PORT_CONTROL.UnconstrainedPower to 1 and TX_SCEDB.SourceInputs[0] to 1.
42	Retimer_SoC_OVR_Force_PWR_Event	Input	When this GPIO is low and there is not a Type-C connection on a port X, then the PD controller drives the Retimer_Force_PWR_Event_PortX GPIO and the Retimer_Reset_N GPIO low. Otherwise, the Retimer_Force_PWR_Event_PortX and Retimer_Reset_N are driven high. See DELAY_CONFIG (register 0x43) section for timing details.
41	ProcHot_N_Event	Output	This recommended to be assigned to a specific GPIO (check data-sheet). This is asserted low when certain power events occur. • When a source or sink is connected. • When the source is disconnected. • Upon entry into a new Explicit Contract as a sink that reduces voltage or current. • When a PR_swap is accepted.
40	Retimer_Reset_N_Event_Port2 [2] [1]	Output	Reset signal for external retimer attached to Port2.
39	Retimer_Reset_N_Event_Port1	Output	Reset signal for external retimer attached to Port1.
38	Retimer_Force_PWR_Event_Port2 [2] [1]	Output	Power control for external retimer attached to Port2.
37	Retimer_Force_PWR_Event_Port1	Output	Power control for external retimer attached to Port1.
36	Fault_Condition_Active_Low_Event_Port2 [2] [1]	Output	Asserts low on an overcurrent event on Port2.
35	Fault_Condition_Active_Low_Event_Port1	Output	Asserts low on an overcurrent event on Port1.
34	Fault_Input_Event_Port2 [2] [1]	Input	When set low by the system, Port2 enters the Type-C Error Recovery State. When set high, no action is taken.
33	Fault_Input_Event_Port1	Input	When set low by the system, Port1 enters the Type-C Error Recovery State. When set high, no action is taken.
32	TBT_Mode_Selection_Event_Port2 [2] [1]	Output	Output: Asserted high when data connection on Port2 is TBT, otherwise low.
31	TBT_Mode_Selection_Event_Port1	Output	Output: Asserted high when data connection on Port1 is TBT, otherwise low.
30	UFP_DFP_Event_Port2 [2] [1]	Output	Output: Asserted high when Port2 is operating as UFP. Asserted low when port is operating as DFP.
29	UFP_DFP_Event_Port1	Output	Output: Asserted high when Port1 is operating as UFP. Asserted low when port is operating as DFP.
28	DP_or_USB3_Event_Port2 [2] [1]	Output	Output: Asserted high when data connection on Port2 is DisplayPort or USB3; Low if neither data mode is active or port is disconnected.
27	DP_or_USB3_Event_Port1	Output	Output: Asserted high when data connection on Port1 is DisplayPort or USB3; Low if neither data mode is active or port is disconnected.
26	User_SVID_Active_Event_Port2 [2] [1]	Output	Asserted high when an Alternate Mode is entered on Port2 using the User Defined SVID.
25	User_SVID_Active_Event_Port1	Output	Asserted high when an Alternate Mode is entered on Port1 using the User Defined SVID.
24	DP_Mode_Selection_Event_Port2 [2] [1]	Output	Output: Asserted high when data connection on Port2 is DP, otherwise low.
23	DP_Mode_Selection_Event_Port1	Output	Output: Asserted high when data connection on Port1 is DP, otherwise low.
22	USB3_Event_Port2 [2] [1]	Output	Output: Asserted high when data connection is USB3 on Port 2, low in all other cases.

Table 3-94. GPIO Events (continued)

Event #	Event Name	I/O	Description
21	USB3_Event_Port1	Output	Output: Asserted high when data connection is USB3 on Port 1, low in all other cases.
20	SourcePDOContractBit2_Port2 [2] [1]	Output	Output: Bit2 of binary encoded outputs indicating when a Source PDO1 through PDO7 on Port2 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired).
19	SourcePDOContractBit1_Port2 [2] [1]	Output	Output: Bit1 of binary encoded outputs indicating when a Source PDO1 through PDO7 on Port2 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired).
18	SourcePDOContractBit0_Port2 [2] [1]	Output	Output: Bit0 of binary encoded outputs indicating when a Source PDO1 through PDO7 on Port2 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired).
17	SourcePDO4Contract_Port2 [2] [1]	Output	Output: Asserted high when a Source PDO4 on Port2 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired), otherwise low.
16	SourcePDO3Contract_Port2 [2] [1]	Output	Output: Asserted high when a Source PDO3 on Port2 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired), otherwise low.
15	SourcePDO2Contract_Port2 [2] [1]	Output	Output: Asserted high when a Source PDO2 on Port2 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired), otherwise low.
14	SourcePDO1Contract_Port2 [2] [1]	Output	Output: Asserted high when a Source PDO1 on Port2 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired), otherwise low.
13	SourcePDOContractBit2_Port1	Output	Output: Bit2 of binary encoded outputs indicating when a Source PDO1 through PDO7 on Port1 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired).
12	SourcePDOContractBit1_Port1	Output	Output: Bit1 of binary encoded outputs indicating when a Source PDO1 through PDO7 on Port1 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired).
11	SourcePDOContractBit0_Port1	Output	Output: Bit0 of binary encoded outputs indicating when a Source PDO1 through PDO7 on Port1 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired).
10	SourcePDO4Contract_Port1	Output	Output: Asserted high when a Source PDO4 on Port1 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired). D-asserted when a PDO other than PDO2 has been negotiated.
9	SourcePDO3Contract_Port1	Output	Output: Asserted high when a Source PDO3 on Port1 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired). D-asserted when a PDO other than PDO2 has been negotiated.
8	SourcePDO2Contract_Port1	Output	Output: Asserted high when a Source PDO2 on Port1 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired). D-asserted when a PDO other than PDO2 has been negotiated.
7	SourcePDO1Contract_Port1	Output	Output: Asserted high when a Source PDO1 on Port1 has been negotiated (the Accept message has been transmitted and the tSrcTransition timer has expired). D-asserted when a PDO other than PDO1 has been negotiated.
6	AMSEL_Event_Port2 [2] [1]	Output	Output: Used to configure external super-speed multiplexors for the DisplayPort Alternate Mode on Port2. Set high when DisplayPort alternate mode is entered and pin assignment A/C/E. Set low when DisplayPort alternate mode is entered and pin assignment B/D/F. Set Hi-Z when DisplayPort alternate mode is not entered.

Table 3-94. GPIO Events (continued)

Event #	Event Name	I/O	Description
5	AMSEL_Event_Port1	Output	Output: Used to configure external super-speed multiplexors for the DisplayPort Alternate Mode on Port1. Set high when DisplayPort alternate mode is entered and pin assignment A/C/E. Set low when DisplayPort alternate mode is entered and pin assignment B/D/F. Set Hi-Z when DisplayPort alternate mode is not entered.
4	Cable_Orientation_Event_Port2[(2)][(1)]	Output	Output: Indicates the plug orientation on Port2. Low when the plug is connected upside-up (CC1 connected to CC in cable) or disconnected. High when plug is connected upside-down (CC2 connected to CC in cable).
3	Cable_Orientation_Event_Port1	Output	Output: Indicates the plug orientation on Port1. Low when the plug is connected upside-up (CC1 connected to CC in cable) or disconnected. High when plug is connected upside-down (CC2 connected to CC in cable).
2	PlugEvent_Port2[(2)][(1)]	Output	Output: Asserted high when plug event has occurred on Port2, otherwise low.
1	PlugEvent_Port1	Output	Output: Asserted high when plug event (attached state) has occurred on Port1, otherwise low.
0	NullEvent	NA	No event associated with this GPIO.

3.41 0x5D RETIMER_DEBUG_MODE Register

After the PD controller writes data from Bytes 3-6 of the 0x50 Data Control register to the Retimer, the PD controller will read back the data it wrote to the Retimer and store that data in this register.

Table 3-95. 0x5D RETIMER_DEBUG_MODE Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x5D	RETIMER_DEBUG_MODE	RW	4	yes	0, cleared on disconnect,

Table 3-96. 0x5D RETIMER_DEBUG_MODE Register Bit Field Definitions

Bits	Name	Description
Bytes 1-4: Retimer Debug Mode Register		
31	TraceMode	
30	Enable	
29:28	DebugModeType	
26:24	AdditionalSkew	
23:22	LaneCount	
21:20	Freq	
19:18	EmphasisDB	
17:16	AmpDB	
15	DebugRxLocked	
14:8	Reserved	
7:6	TPSType	
5:4	RequestedWindowSize	
3:2	EmphasisIcl	
1:0	AmpIcl	

3.42 0x5F DATA_STATUS Register

Table 3-97. 0x5F DATA_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x5F	DATA_STATUS	RO	5	yes	Cleared on disconnect

Table 3-98. 0x5F DATA_STATUS Register Bit Field Definitions

Bits	Name	Description	
Byte 5: Debug Alternate Mode ID			
7:0	DebugAlternateModeID	NIDnT Overlay Number	
Byte 4: Miscellaneous Information			
7:6	Reserved		
5:4	TBTCableGEN	Thunderbolt cable generation.	
		0h	3rd generation TBT (10.3125 and 20.625 Gb/s)
		1h	4th generation TBT (10.0, 10.3125, 20.0 and 20.625 Gb/s)
		2h-3h	Reserved
3:1	TBTCableSpeedSupport		
		0	Reserved
		1h	USB3.1 gen1 cable (10Gb/s Thunderbolt support)
		2h	10Gb/s only
		3h	10Gb/s and 20Gb/s only
		4h-7h	Reserved
0	Reserved		
Byte 3: TBT Information			
7	USB4Connection	Asserted when USB4 mode is entered with the "Host Present" bit asserted.	
6	ActiveCable	Indicates if cable is passive (0) or active (1).	
5	DebugAlternateMode	Asserted when a debug alternate mode is entered.	
4	ActiveLinkTraining	Asserted when the cable is active with uni-directional LSRX communication.	
3	vProDockDetected	Asserted when vPro dock detected.	
2	CableType	Indicates the type of cable: Non-Optical (0) or Optical (1)	
1	TBTType	Indicates the type of Thunderbolt connection: Type-C to Type-C cable (0) or Legacy Adaptor (1)	
0	TBTConnection	Status of Thunderbolt connection.	
		0b	No Thunderbolt connection. This value is also used if TBT Mode is active but an Attention SVDM has been sent/received enabling USB2 instead of TBT.
		1b	Thunderbolt connection present.
Byte 2: DP and Debug Accessory Information			
7	HpdLevel	Status of HPD Level. This is the HPD level received from DisplayPort Sink to TBT controller through the PD controller. This bit is applicable only when the port is acting as a DisplayPort host in virtual HPD signalling mode.	
6	HpdIrqSticky	Status of HPD IRQ event received. This event is from DP Sink to TBT controller through the PD controller. This bit is applicable only when the port is acting as a DisplayPort host in virtual HPD signalling mode.	
5	IrqAck[1][2]		
		0b	No HPD_IRQ_ACK
		1b	HPD_IRQ_ACK
4	DebugAccessoryMode	Asserted when debug accessory is present.	

[1] This feature is not supported by TPS65994AD_F509.04.02.
[2] This feature is not supported by TPS65993AD_F509.04.02.

Table 3-98. 0x5F DATA_STATUS Register Bit Field Definitions (continued)

Bits	Name	Description	
3:2	DPPinAssignment	DisplayPort Pin Assignments	
		00b	Legacy' DP, USB-C to DP cable. Spec pin assignments E-F, if supported.
		01b	Legacy' DP, USB-C to USB-C cable. Spec pin assignments C-D, if supported.
		10b	New' DP, USB-C to USB-C cable. Spec pin assignments A-B, if supported.
		11b	Reserved
1	DPSourceSink	The PD controller does not support DP sink. So this bit always reads back as 0b.	
0	DPConnection	Asserted when there is a DisplayPort connection.	
Byte 1: Connection Information			
7	DataRole	USB data role: DFP (0) or UFP (1).	
6	USB3Speed	USB3 data speed	
		0b	Limited to Gen 1 speed (5Gbps).
		1b	Allowed to Gen 2 speed (10Gbps).
5	USB3Connection	Status of USB3 connection	
		0b	No USB3 connection.
		1b	USB3 connection on SSTx1/Rx1 if upside-up, STx2/Rx2 if upsidedown.
4	USB2Connection	Status of USB2 connection	
		0b	No USB2 connection to USB_RP.
		1b	USB2 connection to USB_RP on 'Mission' D+/D- pair.
3	OvercurrentOrTemperature	Over-current or over-temperature has occurred.	
2	RetimerOrRedriver	Indicates type of active element in the cable.	
		0b	Re-driver.
		1b	Re-timer.
1	ConnectionOrientation	Plug orientation	
		0b	Plug is oriented on CC1 (upside-up). This may also indicate no data connection.
		1b	Plug is oriented on CC2 (upside-down). This value only occurs when there is a valid data connection.
0	DataConnection	Status of data connection.	
		0b	No data connection. The rest of the bits in this register are cleared.
		1b	Data connection present. At least one other bit in this register is non-zero.

Equivalent DisplayPort spec pin assignment mapping:

Table 3-99. 0x5F Data Status (Equivalent DP Specification Pin Assignment Mapping)

DPPinAssignment	USB3Connection	DPSourceSink = 0	DPSourceSink = 1
00	0	DP Source Pin Assignment "E"	DP Sink Pin Assignment "E"
00	1	DP Source Pin Assignment "F"	INVALID
01	0	DP Source Pin Assignment "C"	DP Sink Pin Assignment "C"
01	1	DP Source Pin Assignment "D"	DP Sink Pin Assignment "D"
10	0	DP Source Pin Assignment "A"	DP Sink Pin Assignment "A"
10	1	DP Source Pin Assignment "B"	DP Sink Pin Assignment "B"
11	0	INVALID	INVALID
11	1	INVALID	INVALID

3.43 0x60 RX_USER_SVID_ATTN_VDM Register

NOTE: Only Structured VDM messages for User SVID with Command Type = Initiator (00b) and Command = Attention (6) get stored in this buffer. See register 0x61 () for all other inbound VDMs.

Table 3-100. 0x60 RX_USER_SVID_ATTN_VDM Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x60	RX_USER_SVID_ATTN_VDM	RO	9	yes	Cleared on disconnect/connect/Hard Reset

Table 3-101. 0x60 RX_USER_SVID_ATTN_VDM Register Bit Field Definitions

Bits	Name	Description
Bytes 6-9: Data Object #2		
31:0	RxVdmDo2	Second Data Object of most recently received Attention SVDM. This is the VDO (if present).
Bytes 2-5: Data Object #1		
31:0	RxVdmDo1	VDM header. This field contains the first Data Object of most recently received Attention SVDM.
Byte 1: Status		
7:5	seqNum	Increments by one every time this register is updated, rolls over upon reflow.
4:3	Reserved	
2:0	numValidVDOs	Number of valid VDOs received. Each VDO is 4 bytes. The USB PD spec does not allow more than two VDO's.

3.44 0x61 RX_USER_SVID_OTHER_VDM Register

Table 3-102. 0x61 RX_USER_SVID_OTHER_VDM Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x61	RX_USER_SVID_OTHER_VDM	RO	29	yes	Cleared on disconnect/connect/Hard Reset

Table 3-103. 0x61 RX_USER_SVID_OTHER_VDM Register Bit Field Definitions

Bits	Name	Description	
Bytes 26-29: DO #7 (treated as a 32-bit little endian value)			
31:0	RxVdmDo7		
Bytes 22-25: DO #6 (treated as a 32-bit little endian value)			
31:0	RxVdmDo6	Sixth Data Object of most recently received VDM.	
Bytes 18-21: DO #5 (treated as a 32-bit little endian value)			
31:0	RxVdmDo5	Fifth Data Object of most recently received VDM.	
Bytes 14-17: DO #4 (treated as a 32-bit little endian value)			
31:0	RxVdmDo4	Fourth Data Object of most recently received VDM.	
Bytes 10-13: DO #3 (treated as a 32-bit little endian value)			
31:0	RxVdmDo3	Third Data Object of most recently received VDM.	
Bytes 6-9: DO #2 (treated as a 32-bit little endian value)			
31:0	RxVdmDo2	Second Data Object of most recently received VDM.	
Bytes 2-5: DO #1 (treated as a 32-bit little endian value)			
31:0	RxVdmDo1	First Data Object of most recently received VDM.	
Byte 1: Status			
7:5	seqNum	Increments by one every time this register is updated, rolls over upon reflow.	
4:3	sopType	Frame type of the message in this register.	
		00b	VDM came from SOP.
		01b	VDM came from SOP'.
		10b	VDM came from SOP''.
		11b	VDM came from SOP*_Debug.
2:0	numValidVDOs	Number of DO's received.	

3.45 0x62 BINARYDATA_INDICES Register

This register can only be changed as part of Application Customization. It allows the PD controller to be configured to write specific commands to the I2C3m port when certain events occur. This register also allows for country code information to be customized. The Application Customization tool creates this register automatically.

Table 3-104. 0x62 BINARYDATA_INDICES Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x62	BINARYDATA_INDICES	RO	8	no	Initialized by Application Customization

3.46 0x63 MIPI_VID_STATUS Register

Table 3-105. 0x63 MIPI_VID_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x63	MIPI_VID_STATUS	RO	1	yes	Cleared on disconnect/Hard Reset

Table 3-106. 0x63 MIPI_VID_STATUS Register Bit Field Definitions

Bits	Name	Description
Byte 1: MIPI Mode Status		
7:2	Reserved	
1	DebugModeStatus	This bit is asserted if the Debug Mode has been entered. Otherwise this bit is clear.
0	MipiVidDetected	This bit is asserted if the MIPI VID mode has been entered. Otherwise this bit is clear.

3.47 0x64 I2CMASTER_CONFIG Register

This register allows the PD controller to be configured to write specific commands to specific slave addresses on the I2C3m port. These can be associated with certain events. This register is created automatically by the Application Customization tool.

Table 3-107. 0x64 I2CMASTER_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x64	I2CMASTER_CONFIG	RO	11	yes	0 unless initialized by Application Customization

Table 3-108. 0x64 I2CMASTER_CONFIG Register Bit Field Definitions

Bits	Name	Description
31:28	SClkHighTime[1][2]	For internal/GUI use only. I2C SCLK Hi time cycle See I2CMTPR_REG description. Field location moved to GLOBAL_SYSTEM_CONFIG in ROM2
27:24	SClkLowTime[1][2]	For internal/GUI use only.I2C SCLK low time cycle See I2CMTPR_REG description. Field location moved to GLOBAL_SYSTEM_CONFIG in ROM2
30:23	HSModeEnable[1][2]	For internal/GUI use only. High speed mode enable. When set to 1, SCL clock period can be set upto 3.33Mbps. See I2CMTPR_REG description. Field location moved to GLOBAL_SYSTEM_CONFIG in ROM2
Byte 14: I2CMasterCLKCfg		
23:16	TimePeriod[1][2]	For internal/GUI use only. I2C SCLK Time Period(TPR). See I2CMTPR_REG description. Field location moved to GLOBAL_SYSTEM_CONFIG in ROM2.
Byte 11: AMD Renoir Mux slave address mapping		
7	Reserved	
6:0	SlaveAddrAMDMux	Sets I2C address associated with AMD Renoir Mux. This is only used if PORT_CONFIG.AMDI2CMuxEnable is set to 1.
Byte 10: TBT Retimer 2 slave address mapping (two TBT Retimers in series)		
7	Reserved	
6:0	SlaveAddrTbt2	Thunderbolt retimer slave address for the second Retimer on this port. This is only used if INTEL_VID_CONFIG.DualTBTRetimerPresent is set to 1.
Byte 9: TBT Retimer 1 slave address mapping		
7	Reserved	
6:0	SlaveAddrTbt1	Thunderbolt retimer slave address for the first Retimer on this port. This is only used if INTEL_VID_CONFIG.TBTRetimerPresent is set to 1.
Byte 8: Index 8 slave address mapping		
7	Reserved	
6:0	SlaveAddr8	Sets I2C address associated with events for index 8.
Byte 7: Index 7 slave address mapping		
7	Reserved	
6:0	SlaveAddr7	Sets I2C address associated with events for index 7.
Byte 6: Index 6 slave address mapping		
7	Reserved	
6:0	SlaveAddr6	Sets I2C address associated with events for index 6.
Byte 5: Index 5 slave address mapping		
7	Reserved	
6:0	SlaveAddr5	Sets I2C address associated with events for index 5.
Byte 4: Index 4 slave address mapping		
7	Reserved	
6:0	SlaveAddr4	Sets I2C address associated with events for index 4.
Byte 3: Index 3 slave address mapping		
7	Reserved	

[1] This feature is not supported by TPS65994AD_F509.04.02.

[2] This feature is not supported by TPS65993AD_F509.04.02.

Table 3-108. 0x64 I2CMASTER_CONFIG Register Bit Field Definitions (continued)

Bits	Name	Description
6:0	SlaveAddr3	Sets I2C address associated with events for index 3.
Byte 2: Index 2 slave address mapping		
7	Reserved	
6:0	SlaveAddr2	Sets I2C address associated with events for index 2.
Byte 1: Index 1 slave address mapping		
7	Reserved	
6:0	SlaveAddr1	Sets I2C address associated with events for index 1.

3.48 0x69 TYPEC_STATE Register

Table 3-109. 0x69 TYPEC_STATE Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x69	TYPEC_STATE	RO	4	yes	0

Table 3-110. 0x69 TYPEC_STATE Register Bit Field Definitions

Bits	Name	Description	
31:24	TypeCPortState	Present state of Type-C state-machine.	
		00h	Disabled
		01h-04h	Reserved.
		05h	ErrorRecovery
		06h-23h	Reserved.
		24h	Unattached.Accessory
		25h-2Ah	Reserved.
		2Bh	AttachWait.Accessory
		2Ch-44h	Reserved.
		45h	Try.SRC
		46h-4Dh	Reserved.
		4Eh	TryWait.SNK
		4Fh	Try.SNK
		50h	TryWait.SRC
		51-5Fh	Reserved.
		60h	Attached.SRC
		61h	Attached.SNK
		62h	AudioAccessory
		63h	DebugAccessory
		64h	AttachWait.SRC
		65h	AttachWait.SNK
		66h	Unattached.SNK
		67h	Unattached.SRC
		68h-FFh	Reserved.
23:16	Cc2PinState	State of CC2 pin	
		00h	Not connected
		01h	Ra detected (Source only)
		02h	Rd detected (Source only)
		03h	USB Default Advertisement detected (SInk only)
		04h	1.5A Advertisement detected (Sink Only)
		05h	3.0A Advertisement detected (Sink Only)
		06h-FFh	Reserved
15:8	Cc1PinState	State of CC1 pin	
		00h	Not connected
		01h	Ra detected (Source only)
		02h	Rd detected (Source only)
		03h	USB Default Advertisement detected (SInk only)
		04h	1.5A Advertisement detected (Sink Only)
		05h	3.0A Advertisement detected (Sink Only)
		06h-FFh	Reserved

Table 3-110. 0x69 TYPEC_STATE Register Bit Field Definitions (continued)

Bits	Name	Description	
7:0	CcPinForPd	CC pin used for PD communication.	
		00h	Not connected
		01h	CC1 is used for USB PD communication.
		02h	CC2 is used for USB PD communication.
		03h-FFh	Reserved

3.49 0x6A ADC_RESULTS Register

The PD controller periodically measures the pins mentioned in this register and updates the register accordingly. The frequency of the update depends upon the mode of the PD controller. For example, in Unconnected Sleep the PD controller will not update these registers.

Table 3-111. 0x6A ADC_RESULTS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x6A	ADC_RESULTS	RO	9[1]	no	0

[1] TPS65994AD_F509.04.02 has length of 9 bytes. TPS65993AD_F509.04.02 has length of 9 bytes. TPS65992xAD_F509.05.02 has length of 11 bytes.

Table 3-112. 0x6A ADC_RESULTS Register Bit Field Definitions

Bits	Name	Description
87:80	GPIO2[1][2]	Most recent voltage on the GPIO2 pin. (14mV per LSB)
79:72	GPIO0[1][2]	Most recent voltage on the GPIO0 pin. (14mV per LSB)
71:64	GPIO5	Most recent voltage on the GPIO5 pin. (14mV per LSB)
63:56	GPIO4	Most recent voltage on the GPIO4 pin. (14mV per LSB)
55:48	I_PB_VBUS[2][3]	Most recent current measurement through PP_5V2. (16.5mA per LSB)
47:40	I_PA_VBUS	Most recent current measurement through PP_5V1. (16.5mA per LSB)
39:32	PB_VBUS[2][3]	Most recent voltage on the PB_VBUS pin. (98mV per LSB)
31:24	PA_VBUS	Most recent voltage on the PA_VBUS pin. (98mV per LSB)
23:16	LDO_3V3	Most recent voltage on the LDO_3V3 pin. (14mV per LSB)
15:8	ADCIN2	Most recent voltage on the ADCIN2 pin. (14mV per LSB)
7:0	ADCIN1	Most recent voltage on the ADCIN1 pin. (14mV per LSB)

[1] This feature is not supported by TPS65994AD_F509.04.02.

[2] This feature is not supported by TPS65993AD_F509.04.02.

[3] This feature is not supported by TPS65992xAD_F509.05.02.

3.50 0x6C EVENT_CONFIGURATION Register

This register can only be changed as part of Application Customization. It allows the PD controller to be configured to take specific actions when certain events occur. The Application Customization tool creates this register automatically.

Table 3-113. 0x6C EVENT_CONFIGURATION Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x6C	EVENT_CONFIGURATION	RO	60	yes	0 unless initialized by Application Customization

3.51 0x70 SLEEP_CONFIG Register

Table 3-114. 0x70 SLEEP_CONFIG Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x70	SLEEP_CONFIG	RW	1	no	Initialized by Application Configuration

Table 3-115. 0x70 SLEEP_CONFIG Register Bit Field Definitions

Bits	Name	Description	
7:3	Reserved		
2:1	SleepTime	Minimum time the PD controller waits before entering sleep mode.	
		00b	Reserved
		01b	100 ms
		10b	1200 ms
		11b	Reserved
0	SleepModeAllowed	If this bit is asserted the PD controller will enter sleep modes after device is idle for Sleep Time.	

3.52 0x72 GPIO_STATUS Register

Check the device-specific datasheet for the available GPIO because it may vary by device type.

Table 3-116. 0x72 GPIO_STATUS Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x72	GPIO_STATUS	RO	8	no	N/A

Table 3-117. 0x72 GPIO_STATUS Register Bit Field Definitions

Bits	Name	Description
Bytes 5-8: Direction Configuration		
31:9	Reserved	
8	GPIO8Dir[(1)]	This bit is asserted when this GPIO is configured as an output.
7	GPIO7Dir	This bit is asserted when this GPIO is configured as an output.
6	GPIO6Dir	This bit is asserted when this GPIO is configured as an output.
5	GPIO5Dir	This bit is asserted when this GPIO is configured as an output.
4	GPIO4Dir	This bit is asserted when this GPIO is configured as an output.
3	GPIO3Dir	This bit is asserted when this GPIO is configured as an output.
2	GPIO2Dir	This bit is asserted when this GPIO is configured as an output.
1	GPIO1Dir	This bit is asserted when this GPIO is configured as an output.
0	GPIO0Dir	This bit is asserted when this GPIO is configured as an output.
Bytes 1-4: Output Data		
31:13	Reserved	
12	GPIO12Data	Asserted if a logic high is detected on the GPIO.
11	Reserved	
10	Reserved	
9	Reserved	
8	GPIO8Data[(1)]	Asserted if a logic high is detected on the GPIO.
7	GPIO7Data	Asserted if a logic high is detected on the GPIO.
6	GPIO6Data	Asserted if a logic high is detected on the GPIO.
5	GPIO5Data	Asserted if a logic high is detected on the GPIO.
4	GPIO4Data	Asserted if a logic high is detected on the GPIO.
3	GPIO3Data	Asserted if a logic high is detected on the GPIO.
2	GPIO2Data	Asserted if a logic high is detected on the GPIO.
1	GPIO1Data	Asserted if a logic high is detected on the GPIO.
0	GPIO0Data	Asserted if a logic high is detected on the GPIO.

[(1)] This feature is not supported by TPS65992xAD_F509.05.02.

3.53 0x73 TX_MIDB_SOP Register

The host must enable this feature using the SupportManufacturerInfoMsg bit in the PD3 Configuration register (0x42). If the SupportManufacturerInfoMsg bit is set to 0, then when a Get_Manufacturer_Info message is received the PD controller responds with a Not_Supported message. If the SupportManufacturerInfoMsg bit is set to 1, then the PD controller responds to a Get_Manufacturer_Info message with a target specified as "Port" by pulling the VID and PID from the TX_IDENTITY register (0x47) and appending the contents of this register. If received Get_Manufacturer_Info message has a target specified as "Battery", then the PD controller responds by pulling the VID and PID from the TX_IDENTITY register (0x47) and appending the ASCII string "Not Supported" followed by a zero byte.

Table 3-118. 0x73 TX_MIDB_SOP Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x73	TX_MIDB_SOP	RW	22	no	0 unless initialized by Application Customization

Table 3-119. 0x73 TX_MIDB_SOP Register Bit Field Definitions

Bits	Name	Description
175:0	ManufacturerString	Manufacturer String as defined in USB PD. This must be a null terminated string. The PD controller always sends all 22 bytes.

3.54 0x74 RX_ADO Register

Table 3-120. 0x74 RX_ADO Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x74	RX_ADO	RO	4	yes	Reset upon disconnect/Hard Reset,,

Table 3-121. 0x74 RX_ADO Register Bit Field Definitions

Bits	Name	Description
Bytes 1-4: Most recently received Alert Data Object (ADO)		
31:24	TypeOfAlert	Type of alert as defined by USB PD.
23:20	FixedBatteries	Status of fixed batteries when selected by AlertType.
19:16	HotSwappableBatteries	Status of hot swappable batteries when selected by AlertType
15:0	Reserved	

3.55 0x75 TX_ADO Register

When the host issues the 4CC command 'ALRT' the contents of this register are used to send the Alert message. Note that this register is not duplicated for each port. The host should use the following sequence to send the same Alert message on both ports:

- Write Transmit Alert Data Object (0x75)
- Issue the 4CC 'ALRT' command to Port 1.
- Issue the 4CC 'ALRT' command to Port 2.

The host should use the following sequence to send two different Alert messages on each port (only relevant for dual-port PD controllers):

- Write Transmit Alert Data Object (0x75)
- Issue the 4CC 'ALRT' command to Port 1.
- Wait for the command to complete.
- Write Transmit Alert Data Object (0x75)
- Issue the 4CC 'ALRT' command to Port 2.

Table 3-122. 0x75 TX_ADO Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x75	TX_ADO	RW	4	no	0

Table 3-123. 0x75 TX_ADO Register Bit Field Definitions

Bits	Name	Description
Bytes 1-4: Alert Data Object (ADO) to be transmitted		
31:24	TypeOfAlert	Type of alert as defined by USB PD.
23:20	FixedBatteries	Status of fixed batteries when selected by AlertType.
19:16	HotSwappableBatteries	Status of hot swappable batteries when selected by AlertType
15:0	Reserved	

3.56 0x77 TX_SCEDB Register

If the PD3 configuration register (0x42) bit SourceCapExtMsg is set to zero, the PD controller responds to a Get_Source_Cap_Extended USB PD message with a Not_Supported message. If the SourceCapExtMsg bit is set to 1 then the response is generated from the contents of this register based on USB PD requirements. The VID, PID, and XID fields are taken from the TX_IDENTITY Register (0x47), the FW version is taken from the RR word in the Version register (0x0F), the HW version is taken from the REV_ID word in the Boot Flags register (0x2D), then the contents of this register are appended.

Table 3-124. 0x77 TX_SCEDB Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x77	TX_SCEDB	RW	14	no	Initialized by Application Configuration

Table 3-125. 0x77 TX_SCEDB Register Bit Field Definitions

Bits	Name	Description
111	Reserved	
110:104	SourcePDP	Source's PDP rating as defined by the USB PD specification.
103:100	NumHotSwappableBatteries	Number of hot swappable batteries / battery slots as defined by the USB PD specification. (Max of 1)
99:96	NumFixedBatteries	Number of fixed batteries / battery slots as defined by the USB PD specification. (Max of 3)
95:88	SourceInputs	Source inputs as defined by the USB PD specification. The Barrel_Jack_Event GPIO Event can modify bit 0 of this field automatically upon rising and falling edges. The host may choose to set bit 1 of this field to 1 when it enables the Barrel_Jack_Event GPIO Event.
87:80	TouchTemp	Touch temperature as defined by the USB PD specification.
79:64	PeakCurrent3	Peak Current 3 as defined by the USB PD specification.
63:48	PeakCurrent2	Peak Current 2 as defined by the USB PD specification.
47:32	PeakCurrent1	Peak Current 1 as defined by the USB PD specification.
31:24	TouchCurrent	Touch current as defined by the USB PD specification.
23:16	Compliance	Compliance as defined by the USB PD specification.
15:8	HoldupTime	Hold up time as defined by the USB PD specification.
7:0	VoltageRegulation	Voltage regulation as defined by the USB PD specification.

3.57 0x79 TX_SDB Register

This feature must be enabled in the PD3 configuration register (0x42) SupportStatusMsg bit. The PD controller does not take any automatic action if this register is written.

If the SupportStatusMsg bit set to 0 and a Get_Status message is received, then the PD controller ignores this TX_SDB register and responds with a Not_Supported message. If the SupportStatusMsg bit is set to 1 and a Get_Status message is received, then the response is determined as described below.

The Power Status byte is partially computed by the PD controller. Specifically, there are three bits the PD controller may compute:

- bit 1: The PD controller will send a Discover Identity message to the cable and if the cable reports it can handle less than the PD controller would otherwise advertise (based on Tx Source Capabilities register (0x32)) then bit1 is asserted. See Table below for cable maximum current capability determination.
- bit 2: If the host has enabled GPIO Event #80 (Prevent_High_Current_Contract_Event) then bit2 is computed as high if the associated GPIO has a low value.
- bit 3: The PD controller computes bit3 as the value of the dead-battery flag.

When a Get_Status message is received by the PD controller then it formulates the response as the first 5 bytes of this register then appending the bit-wise OR of the computed value of bits 1, 2, and 3 with the Power Status byte written by the host. Note that the Power Status byte in this register will always read back as the value written by the host.

The host should update this register when it updates the Tx Source Capabilities register (0x32).

The PD controller does not take any automatic action if this register is written. The host is responsible for issuing 4CC 'ALRT' command to inform the Port Partner that the Status is changed.

Table 3-126. 0x79 TX_SDB Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x79	TX_SDB	RW	6	yes	0

Table 3-127. 0x79 TX_SDB Register Bit Field Definitions

Bits	Name	Description
47:40	PowerStatus	Power status as defined by the USB PD specification.
39:32	TemperatureStatus	Temperature status as defined by the USB PD specification.
31:24	EventFlags	Event flags as defined by the USB PD specification.
23:16	PresentBatteryInput	Present battery input as defined by the USB PD specification.
15:8	PresentInput	Present input as defined by the USB PD specification.
7:0	InternalTemp	Internal temperature as defined by the USB PD specification.

Table 3-128. Maximum Current Determination.

Active Cable VDO1 Information		Active Cable VDO2 information		Maximum current to advertise	Comment
VBUS Current Handling (bits 6:5)	VBUS through cable (bit 4)	SuperSpeed supported (bit 4)	SuperSpeed Lanes Supported (bit 3)		
X	0	X	X	No Limit	0
00b	1	0	X	0.5 A	Limit to USB 2.0
00b	1	1	0	0.9 A	Limit to USB 3.1
00b	1	1	1	1.5 A	Limit to USB 3.2
01b	1	X	X	3 A	
10b	1	X	X	5 A	
11b	1	X	X	No Limit	

3.58 0x7B TX_BSDO Register

The host should also program the Tx Source Capabilities Extended register (0x77) in order to specify the number of each type of battery such that it is consistent with the contents of this register. This feature must be enabled in the PD3_CONFIG register (0x42) SupportBatteryStatusMsg bit. The PD controller does not take any automatic action if this register is written. The host is responsible for issuing 4CC 'ALRT' command to inform the Port Partner that the Battery Status is changed.

If the SupportBatteryStatusMsg bit is set to 0 and a Get_Battery_Status message is received, then this register is ignored and the PD controller sends a Not_Supported message. If the SupportBatteryStatusMsg bit is set to 1, and a Get_Battery_Status message is received then the contents of this register are sent in response.

Table 3-129. 0x7B TX_BSDO Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x7B	TX_BSDO	RW	16	no	0

Table 3-130. 0x7B TX_BSDO Register Bit Field Definitions

Bits	Name	Description
Bytes 13-16: Battery status data object returned for hot-swappable battery index 0.		
31:16	HotSwappableBattery0_PresentCapacity	Battery status data object returned for fixed battery index 0.
15:8	HotSwappableBattery0_BatteryInfo	Battery status data object returned for hot-swappable battery index 0.
7:0	Reserved	Battery status data object returned for fixed battery index 0.
Bytes 9-12: Battery status data object returned for fixed battery index 2.		
31:16	FixedBattery2_PresentCapacity	Battery status data object returned for fixed battery index 0.
15:8	FixedBattery2_BatteryInfo	Battery status data object returned for fixed battery index 2.
7:0	Reserved	Battery status data object returned for fixed battery index 0.
Bytes 5-8: Battery status data object returned for fixed battery index 1.		
31:16	FixedBattery1_PresentCapacity	Battery status data object returned for fixed battery index 0.
15:8	FixedBattery1_BatteryInfo	Battery status data object returned for fixed battery index 1.
7:0	Reserved	Battery status data object returned for fixed battery index 0.
Bytes 1-4: Battery status data object returned for fixed battery index 0.		
31:16	FixedBattery0_PresentCapacity	Battery status data object returned for fixed battery index 0.
15:8	FixedBattery0_BatteryInfo	Battery status data object returned for fixed battery index 0.
7:0	Reserved	Battery status data object returned for fixed battery index 0.

3.59 0x7D TX_BCDB Register

The host should also program the Tx Source Capabilities Extended register (0x77) in order to specify the number of each type of battery such that it is consistent with the contents of this register. This feature must be enabled in the PD3_CONFIG register (0x42) bit SupportBatteryCapMsg. The PD controller does not take any automatic action if this register is written.

If the SupportBatteryCapMsg is 0 and a Get_Battery_Capabilities message is received, then the contents of this register are ignored and the PD controller sends a Not_Supported message. If the SupportBatteryCapMsg is 1 and a Get_Battery_Capabilities message is received, then the contents of this register are sent in response.

Table 3-131. 0x7D TX_BCDB Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x7D	TX_BCDB	RW	36	no	0 unless initialized by Application Customization

Table 3-132. 0x7D TX_BCDB Register Bit Field Definitions

Bits	Name	Description
Bytes 28-36: Hot-Swappable Battery 3		
71:64	BatteryType3	Battery type for hot-swappable battery index 0.
63:48	BatteryLastFullChargeCapacity3	Battery last full charge capacity for hot-swappable battery index 0.
47:32	BatteryDesignCapacity3	Battery design capacity for hot-swappable battery index 0.
31:16	Pid3	PID for hot-swappable battery index 0.
15:0	Vid3	VID for hot-swappable battery index 0.
Bytes 19-27: Fixed Battery 2		
71:64	BatteryType2	Battery type for fixed battery index 2.
63:48	BatteryLastFullChargeCapacity2	Battery last full charge capacity for fixed battery index 2.
47:32	BatteryDesignCapacity2	Battery design capacity for fixed battery index 2.
31:16	Pid2	PID for fixed battery index 2.
15:0	Vid2	VID for fixed battery index 2.
Bytes 10-18: Fixed Battery 1		
71:64	BatteryType1	Battery type for fixed battery index 1.
63:48	BatteryLastFullChargeCapacity1	Battery last full charge capacity for fixed battery index 1.
47:32	BatteryDesignCapacity1	Battery design capacity for fixed battery index 1.
31:16	Pid1	PID for fixed battery index 1.
15:0	Vid1	VID for fixed battery index 1.
Bytes 1-9: Fixed Battery 0		
71:64	BatteryType0	Battery type for fixed battery index 0.
63:48	BatteryLastFullChargeCapacity0	Battery last full charge capacity for fixed battery index 0.
47:32	BatteryDesignCapacity0	Battery design capacity for fixed battery index 0.
31:16	Pid0	PID for fixed battery index 0.
15:0	Vid0	VID for fixed battery index 0.

3.60 0x7E TX_SKEDB Register

This feature must be enabled in the PD3_CONFIG register (0x42) bit SupportSinkCapExtended. The PD controller does not take any automatic action if this register is written.

If the SupportSinkCapExtended bit is 0 and a Get_Sink_Cap_Extended message is received, then the contents of this register are ignored and the PD controller sends a Not_Supported message. If the SupportSinkCapExtended bit is 1 and a Get_Sink_Cap_Extended message is received, then the contents of this register are used to formulate the response. The PD controller also pulls the VID, PID, XID from the TX_IDENTITY register (0x47), the FW version is taken from the RR word in the Version register (0x0F), the HW version is taken from the REV_ID word in the Boot Flags register (0x2D). Finally, the PD controller appends the contents of this register.

Refer to the latest USB PD specification for detailed description of each field. The values in this register are not used by the PD controller to affect behavior, it just simply uses these contents to respond.

Table 3-133. 0x7E TX_SKEDB Register

Address	Name	Access	Length	Unique per Port	Power-Up Default
0x7E	TX_SKEDB	RW	11	no	Initialized by Application Configuration

Table 3-134. 0x7E TX_SKEDB Register Bit Field Definitions

Bits	Name	Description
87:80	SinkMaximumPDP	Sink maximum PDP as defined in the USB PD specification.
79:72	SinkOperationalPDP	Sink operational PDP as defined in the USB PD specification.
71:64	SinkMinimumPDP	Sink minimum PDP as defined in the USB PD specification.
63:56	SinkModes	Sink modes as defined in the USB PD specification.
55:48	BatteryInfo	Battery information as defined in the USB PD specification.
47:40	TouchTemp	Touch temperature as defined by the USB PD specification.
39:32	Compliance	Compliance as defined by the USB PD specification.
31:16	SinkLoadChar	Sink load characteristics as defined in the USB PD specification.
15:8	LoadStep	Load step as defined in the USB PD specification.
7:0	SKEDBVersion	SKEDB Version as defined in the USB PD specification.

Chapter 4

SLVUBT3–May 2020

4CC Task Detailed Descriptions

4.1 Overview

This section describes the 4CC Tasks defined by the PD Controller Host Interface. The Tasks are categorized into various sub-groups in this section. All Tasks that return data using the DATAx registers will always ensure the proper output data is loaded into those registers before setting the CMDx register to 0 to indicate Task completion. DATAx is never modified by PD Controller once CMDx has been changed to 0, to ensure the Host can retrieve data from the previously-executed Task, and to ensure the Host can load these registers for a future Task without risk of overwriting. Note that other registers may continue to be updated after a Task completes, as Tasks may have additional side effects.

Many of the Tasks return a status code in the first byte of the DATAx register. The standard Task response byte is defined in Table 4-1. The remaining DATAx bytes may be used at each Task's discretion.

Table 4-1. Standard Task Response

Description	**Tasks are a special form of Tasks that return a status code in the first byte of the DATAX register.**			
Output DATAX	Bit	Name	Description	
	Byte 1: Task Return Code			
	7:4	Reserved	Reserved for standard Tasks. May be used by certain Tasks for Task-specific return codes. Successful return codes may use this byte provided TaskResult is 0x0.	
	3:0	TaskResult	Standard Task return codes.	
			0x0	Task completed successfully.
			0x1	Task timed-out or aborted by 'ABRT' Request.
			0x2	Reserved.
			0x3	Task rejected.
			0x4	Task rejected because the Rx Buffer was locked. This is for Tasks that would require the PD controller to use the Rx Buffer.
			0x5-0xF	Reserved for standard Tasks. May be used by certain Tasks for Task-specific error codes. Treated as an error when encountered.

4.2 CPU Control Tasks

4.2.1 'Gaid' - Return to normal operation

Table 4-2. 'Gaid' - Return to normal operation.

Description	The 'Gaid' Task causes a warm restart of the PD Controller processor.
INPUT DATAX	None
OUTPUT DATAX	None
Task Completion	Technically this Task never completes since the processor restarts. However, since all HI registers return to their default state upon reboot, all CMDx/DATAx registers will return to 0, which will mark this Task as complete.
Side Effects	The 'Gaid' Task causes a warm restart of the PD Controller processor. PD Controller may momentarily NAK I2C transactions while rebooting.
Additional Information	The PD controller is in the 'APP ' mode, then it immediately goes to the Error Recovery state then after delaying 1 second (typical) it does a warm restart.

4.2.2 'GAID' - Cold reset request

Table 4-3. 'GAID' - Cold reset request

Description	The 'GAID Task causes a cold restart of the PD Controller processor.
INPUT DATAX	None
OUTPUT DATAX	None
Task Completion	Technically this Task never completes since the processor restarts. However, since all HI registers return to their default state upon reboot, all CMDx/DATAx registers will return to 0, which will mark this Task as complete. This Task forces the PD Controller to reboot its OTP bootloader.
Side Effects	The 'Gaid' Task causes a warm restart of the PD Controller processor. PD Controller may momentarily NAK I2C transactions while rebooting.
Additional Information	The PD controller immediately goes to the Error Recovery state, then after delaying 1 second (typical) it does a cold restart.

4.3 Modal Tasks

The following Tasks are considered "Modal" for the PD Controller. Only one Modal Task can be active at a time, if the MODE register (0x03) reports any value other than 'APP ' when a Modal Task is issued the Task will be Rejected unless this Modal Task is considered a higher-priority, in which case it may cancel other Modal Tasks. Modal Tasks change the value of the Mode register as described below, and this value remains unless the Task defines its own mechanism for disabling the mode, or if a 'Gaid'/'GAID' Task is issued to reboot PD Controller and clear the Mode.

4.3.1 'DISC' - Simulate port disconnect

Table 4-4. 'DISC' - Simulate port disconnect

Description	**The 'DISC' Modal Task causes the PD Controller to act as if the USB-C port is disconnected, with an optional Host-specified delay to restoring normal port operation. If currently there is no USB-C connection on the port, then this task will be rejected. The port that will be disconnected when writing the 'DISC' Task will correspond to the I2C slave address used when writing the 'DISC' Task.**		
INPUT DATAX	**Bit**	**Name**	**Description**
	7:0	DISCdelay	8-bit value in seconds for disconnect time. If 0, there is no automatic reconnect.
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	This Task always completes successfully, it has no reason to be rejected or timed-out. If another Modal Task was already active the 'DISC' Modal Task will cancel that Modal Task and take its place. The 'DISC' Modal Task completes immediately, it does not wait for the re-connect delay		
Side Effects	Effectively the action of this Modal Task is to force the Type-C state machine into the Disabled state, which disables the CC pull-up/downs. Any power switch that was enabled as input or output will be disabled either as a direct result of this Modal Task or an indirect result due to the disconnect event. This causes any existing connection to be lost, and the HI registers will be updated as appropriate for a disconnect event. No new connections will be detected because the port is in the Disabled state. The Type-C state machine does not return to normal operation unless DISCdelay is non-zero and the specified delay passes or a 'Gaid'/'GAID' Task is issued to reboot PD Controller .The MODE register (0x03) is changed to 'DIS#' (where # is the port number that is being disconnected) while this Modal Task is active to indicate that PD Controller is not in its normal operating state. If DISCdelay is non-zero when the specified delay expires the Mode register will return to 'APP '.If another Modal Task is enabled that cancels this one then the port shall be re-enabled at that time, regardless of DISCdelay and even if the timer had not yet expired. NOTE: If a hot-VBUS Source is attached VBUS may still be present at the input to PD Controller even though it is in the Disabled state. Some VBUS-related registers may report this voltage presence as usual, however PD Controller will keep its power switches disabled until normal operation is restored.		
Additional Information	None		

4.4 PD Message Tasks

4.4.1 'SWSk' - PD PR_Swap to Sink

Table 4-5. 'SWSk' - PD PR_Swap to Sink

Description	The 'SWSk' Task instructs PD Controller to attempt to become a Sink via PR_Swap at the first opportunity while maintaining policy engine compliance.
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The 'SWSk' Task completes either when the PR_Swap is finished or it is otherwise determined to not be possible or fails. The Task may continue to run because of Wait messages being sent by the Source. The 'SWSk' Task shall be considered rejected if: • The Source indicated via Source Capabilities that it does not support Dual-Role Power. • The PR_Swap is Rejected. The 'SWSk' Task shall be considered timed-out if: • The PR_Swap is Accepted but failed to complete per the PD spec. The 'SWSk' Task shall be considered successful if: • PD Controller is already in the Sink power role. • The PR_Swap is Accepted and completes normally.
Side Effects	When the 'SWSk' Task completes successfully PD Controller will have transitioned to the Sink power role, which impacts other registers. If the PR_Swap fails after the Accept is sent then Soft and/or Hard Resets are likely to occur per PD spec requirements.
Additional Information	None

4.4.2 'SWSr' - PD PR_Swap to Source

Table 4-6. 'SWSr' - PD PR_Swap to Source

Description	The 'SWSr' Task instructs PD Controller to attempt to become a Source via PR_Swap at the first opportunity while maintaining policy engine compliance.
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The 'SWSr' Task completes either when the PR_Swap is finished or it is otherwise determined to not be possible or fails. The Task may continue to run because of Wait messages being sent by the Sink. The 'SWSr' Task shall be considered rejected if: • The Sink previously indicated via Sink or Source Capabilities that it does not support Dual-Role Power. • The PR_Swap is Rejected. The 'SWSr' Task shall be considered timed-out if: • The PR_Swap is Accepted but failed to complete per the PD spec. The 'SWSr' Task shall be considered successful if: • PD Controller is already in the Source power role. • The PR_Swap is Accepted and completes normally.
Side Effects	When the 'SWSr' Task completes successfully PD Controller will have transitioned to the Source power role, which impacts other registers. If the PR_Swap fails after the Accept is sent then Soft and/or Hard Resets are likely to occur per PD spec requirements.
Additional Information	None

4.4.3 'SWDF' - PD DR_Swap to DFP

Table 4-7. 'SWDF' - PD DR_Swap to DFP

Description	**The 'SWDF' Task instructs PD Controller to attempt to become a DFPvia DR_Swap at the first opportunity while maintaining policy enginecompliance. If there are any active Alternate Modes as a UFP PD Controller willattempt to exit those Modes first before sending the DR_Swap.**
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The 'SWDF' Task completes either when the DR_Swap is finished or it is otherwise determined to not be possible or fails. The Task may continue to run because of Wait messages being sent by the DFP. The 'SWDF' Task shall be considered rejected if: • The UFP indicated via Source or Sink Capabilities that it does not support Data Role Swap. • The DR_Swap is Rejected. The 'SWDF' Task shall be considered successful if: • PD Controller is already in the DFP data role. • The DR_Swap is Accepted and completes normally.
Side Effects	When the 'SWDF' Task completes successfully PD Controller will have transitioned to the DFP data role, which impacts other registers. If the DR_Swap fails after the Accept is sent then Soft and/or Hard Resets are likely to occur per PD spec requirements.
Additional Information	None

4.4.4 'SWUF' - PD DR_Swap to UFP

Table 4-8. 'SWUF' - PD DR_Swap to UFP

Description	**The 'SWUF' Task instructs PD Controller to attempt to become a UFPvia DR_Swap at the first opportunity while maintaining policy enginecompliance. If there are any active Alternate Modes as a DFP PD Controller willexit those Modes first before attempting the DR_Swap.**
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The 'SWUF' Task completes either when the DR_Swap is finished or it is otherwise determined to not be possible or fails. The Task may continue to run because of Wait messages being sent by the UFP. The 'SWUF' Task shall be considered rejected if: • The DFP indicated via Source or Sink Capabilities that it does not support Data Role Swap. • The DR_Swap is Rejected. The 'SWUF' Task shall be considered successful if: • PD Controller is already in the UFP data role. • The DR_Swap is Accepted and completes normally.
Side Effects	When the 'SWDF' Task completes successfully PD Controller will have transitioned to the UFP data role, which impacts other registers. If the DR_Swap fails after the Accept is sent then Soft and/or Hard Resets are likely to occur per PD spec requirements.
Additional Information	None

4.4.5 'GSkC' - PD Get Sink Capabilities

Table 4-9. 'GSkC' - PD Get Sink Capabilities

Description	The 'GSkC' Task instructs PD Controller to issue a Get_Sink_Capmessage to the Port Partner at the first opportunity while maintaining policyengine compliance.
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The 'GSkC' Task completes either when the Sink Capabilities message is received or the Task otherwise fails. • The Port Partner is a Source and indicated it was not Dual-Role Power. • The Port Partner responds to the Get_Sink_Cap message with a Reject or Not_Supported message. The 'GSkC' Task shall be considered timed-out if: • The Port Partner fails to respond within the time required by the PD spec. The 'GSkC' Task shall be considered successful if: • The Get_Sink_Cap message is sent, GoodCRC'ed and a Sink Capabilities response is received and processed.
Side Effects	When the 'GSkC' Task completes successfully the *RX_SINK_CAPS* register (0x31) will have been updated.
Additional Information	None

4.4.6 'GSrC' - PD Get Source Capabilities

Table 4-10. 'GSrC' - PD Get Source Capabilities

Description	The 'GSrC' Task instructs PD Controller to issue a Get_Source_Capmessage to the Port Partner device at the first opportunity while maintainingpolicy engine compliance.
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The 'GSrC' Task completes either when the Source Capabilities message is received or the Task otherwise fails. The 'GSrC' Task shall be considered rejected if: • The Port Partner is a Sink and indicated (via previous Source or Sink Capabilities) it was not Dual-Role Power. • The Port Partner responds to the Get_Source_Cap message with a Reject or Not_Supported message. The 'GSrC' Task shall be considered timed-out if: • The Port Partner fails to respond within the time required by the PD spec. The 'GSrC' Task shall be considered successful if: • The Get_Source_Cap message is sent, GoodCRC'ed and a Source Capabilities response is received and processed.
Side Effects	When the 'GSrC' Task completes successfully the *RX_SOURCE_CAPS* register (0x30) will have been updated.
Additional Information	None

4.4.7 'GPPI' - PD Get Port Partner Information

The 'GPPI' Task can be used to cause the PD controller to issue these types of USB PD Get messages:

- Get_Source_Cap_Extended (control message)
- Get_Sink_Cap_Extended (control message)
- Get_Status (Control message)
- Get_Country_Codes (Control message)
- Get_Country_Info (Data message)
- Get_Battery_Status (Extended message)
- Get_Battery_Cap (Extended message)
- Get_Manufacturer_Info (Extended message)

The PD controller does not have dedicated registers to store the response to these messages. The host must get that response from the DATAX register associated with this Task.

The host should NOT use 'GPPI' to send Get_Sink_Capabilities or Get_Source_Capabilities messages, because the USB PD spec requires specific actions be taken by the PD controller any time those messages are received. While executing the 'GPPI' Task, the PD controller does not parse the returned message to carry out those checks. Instead, the host should use 'GSkC' to send Get_Sink_Capabilities and 'GSrC' to send Get_Source_Capabilities messages.

This Task is defined to enable supporting any new Get message that may be defined by USB PD in the future.

Table 4-11. 'GPPI' - Send a USB PD Get* message.

Description	**The 'GPPI' Task instructs PD Controller to issue a specific USB PDmessage to the Port Partner at the first opportunity while maintaining policyengine compliance.**			
INPUT DATAX	**Bit**	**Name**	**Description**	
	15	Reserved		
	14:13	FrameType	00b	SOP
			01b	SOP'
			10b	SOP''
			11b	Reserved
	12:8	NumBytes		
	7	Reserved		
	6:5	MessageCategory	00b	Control message (no payload)
			01b	Data message (requires payload)
			10b	Extended message (requires payload)
			11b	Reserved
	4:0	MessageType	This field should be the MessageType as defined in the USB PD specification. It specifies the Type of message the PD controller will send.	
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.			

Table 4-11. 'GPPI' - Send a USB PD Get* message. (continued)

Description	**The 'GPPI' Task instructs PD Controller to issue a specific USB PDmessage to the Port Partner at the first opportunity while maintaining policyengine compliance.**
Task Completion	The 'GPPI' Task completes either when the appropriate message is received or the Task otherwise fails. The 'GPPI' Task shall be considered rejected if: • Sending the requested message would violate the USB PD spec. For example, the Port Partner is a Sink and indicated (via previous Source or Sink Capabilities) it was not Dual-Role Power. • The PortPartner replies with a Reject or Not_Supported message. • The USB PD spec revision (PlugPartnerNegSpecRev or PortPartnerNegSpecRev in PD3_STATUS register (0x41) does not allow sending the requested message. The 'GPPI' Task shall be considered timed-out if: • The requested message is sent, GoodCRC'ed and the recipient (Port Partner or Cable Plug) fails to respond within the time required by the PD spec. • A PD Hard Reset or a disconnection happens before the Task completes. The 'GPPI' Task shall be considered successful if: • The requested message is sent, GoodCRC'ed and an appropriate response is received and processed. The 'GPPI' Task shall be aborted when the Rx Buffer is locked. The Rx Buffer is locked after data from a receive message is placed in the DATAX register. The Rx Buffer is unlocked after disconnect and by the 'MBRd' Task.
Side Effects	If necessary, the PD controller may issue a VCONN_Swap in order to send the requested message to a Cable Plug. If the PD controller is in the sink power role and it reads Rp = SinkTxNG, it will wait until Rp = SinkTxOK before initiating the atomic message sequence requested by this 'GPPI' Task. This can cause an non-deterministic delay in completing the Task.
Additional Information	The PD controller will continue trying to execute this Task until it times out or aborts as described above. The host may want to issue the 'ABRT' Task if the process takes too long. Some scenarios where this could happen are: • The PD controller is required to be the VCONN_Source in order to send any message on SOP or SOP'. The PD controller will continue trying to become the VCONN provider until it is successful. • The PD controller with a sink power role (i.e. PresentRole = Sink) is required to wait for Rp = SinkTxOK before initiating an Atomic Message Sequence. The PD controller will continue waiting for Rp = SinkTxOK until it is able to send the appropriate message required for this 'GPPI' Task. The host should wait until CMDx reads as 0 or INT_EVENT1.CmdComplete is asserted before issuing the 'MBRd' 4CC Task to read the Rx Buffer after issuing this 'GPPI' Task. While executing the 'GPPI' Task, the PD controller uses the same shared buffer that is used to store other extended messages. Therefore, the host should not use the 'GPPI' Task when any other atomic message sequence is ongoing. To read the PD response received as a result of issuing the 'GPPI' Task once it is completed, the host should use the 'MBRd' 4CC command. The 'MBRd' Task must also be used to unlock the Rx Buffer for other incoming message.

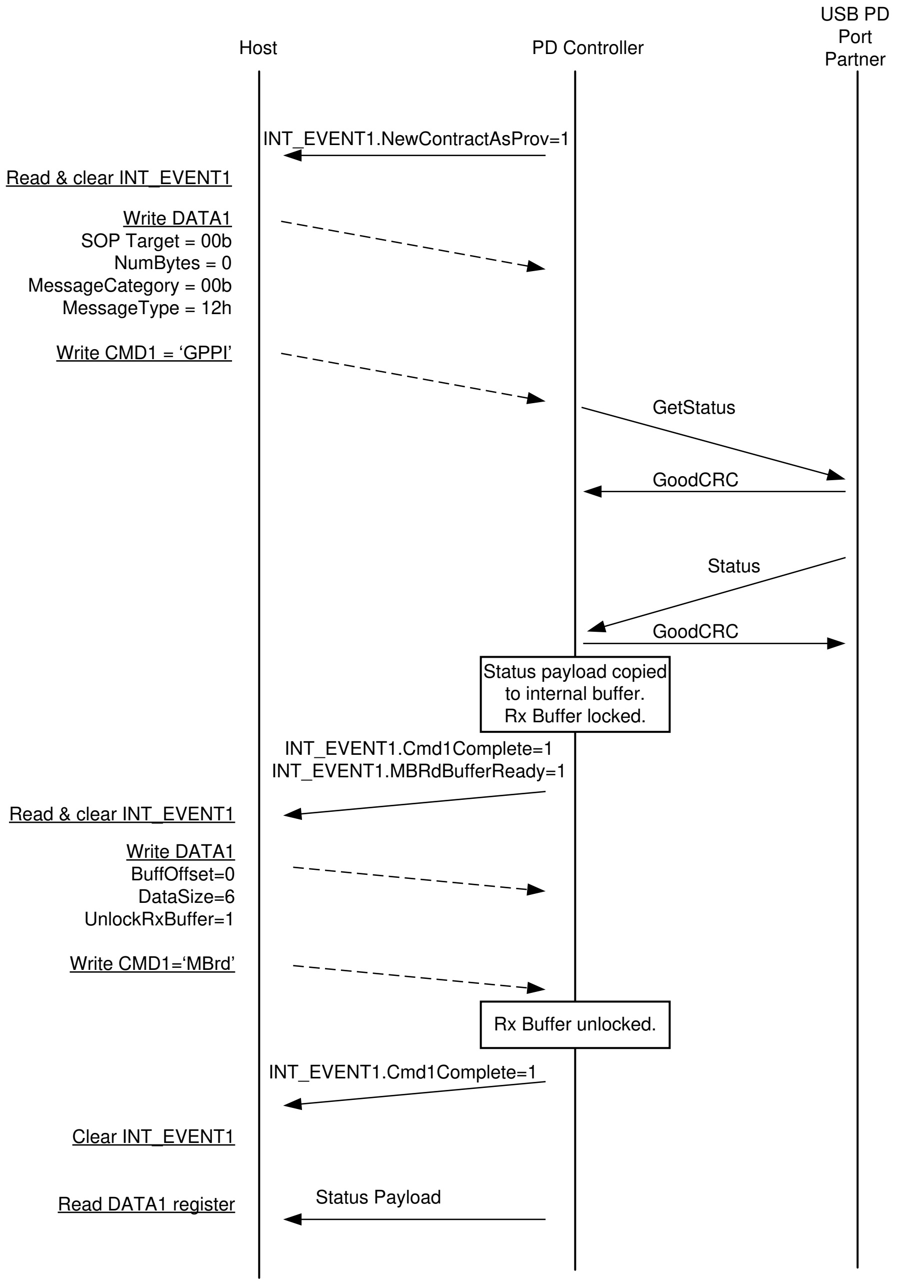


Figure 4-1. Example sequence for 'GPPI' Task when host uses INT_EVENT1.

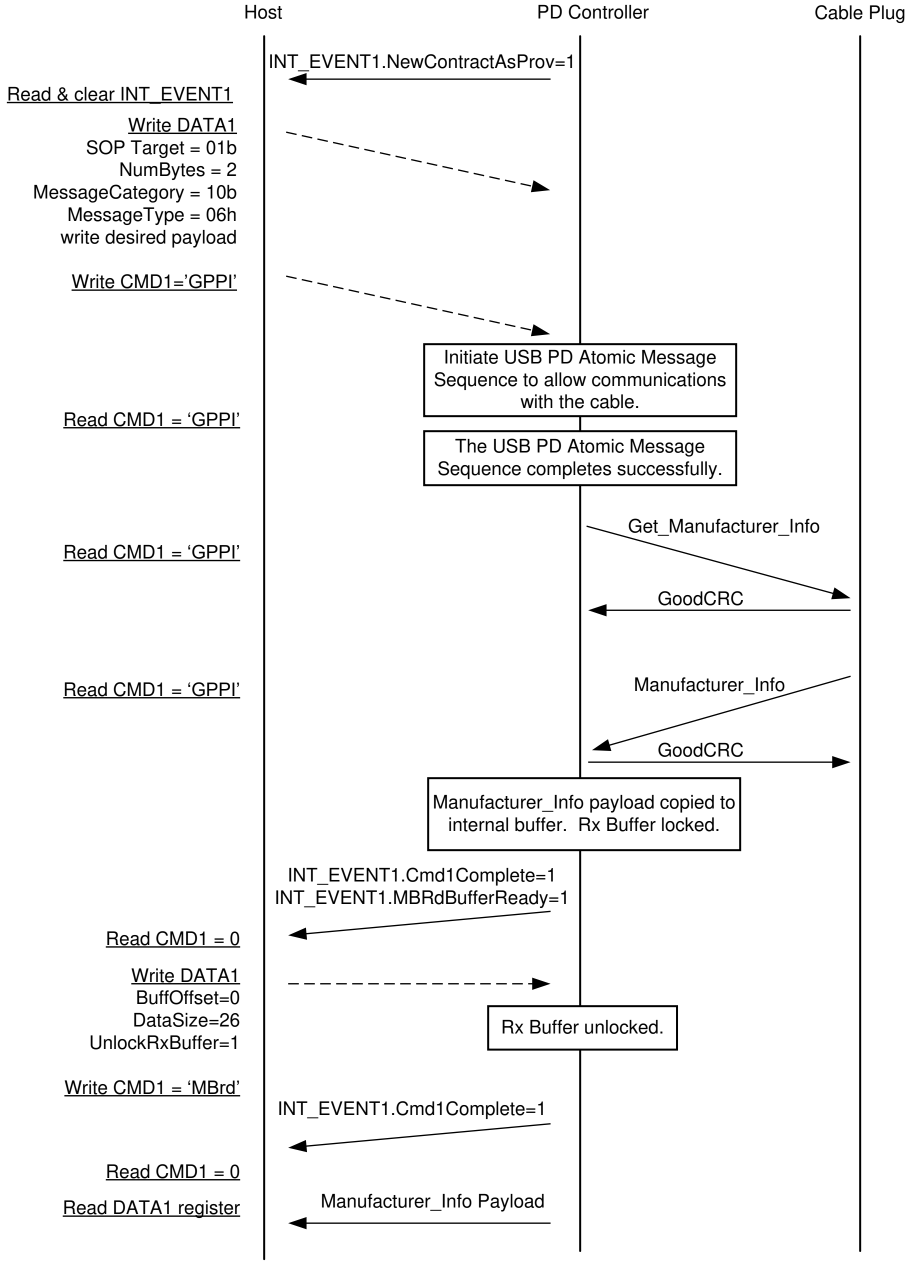


Figure 4-2. Example sequence for 'GPPI' Task when host uses CMD1 polling.

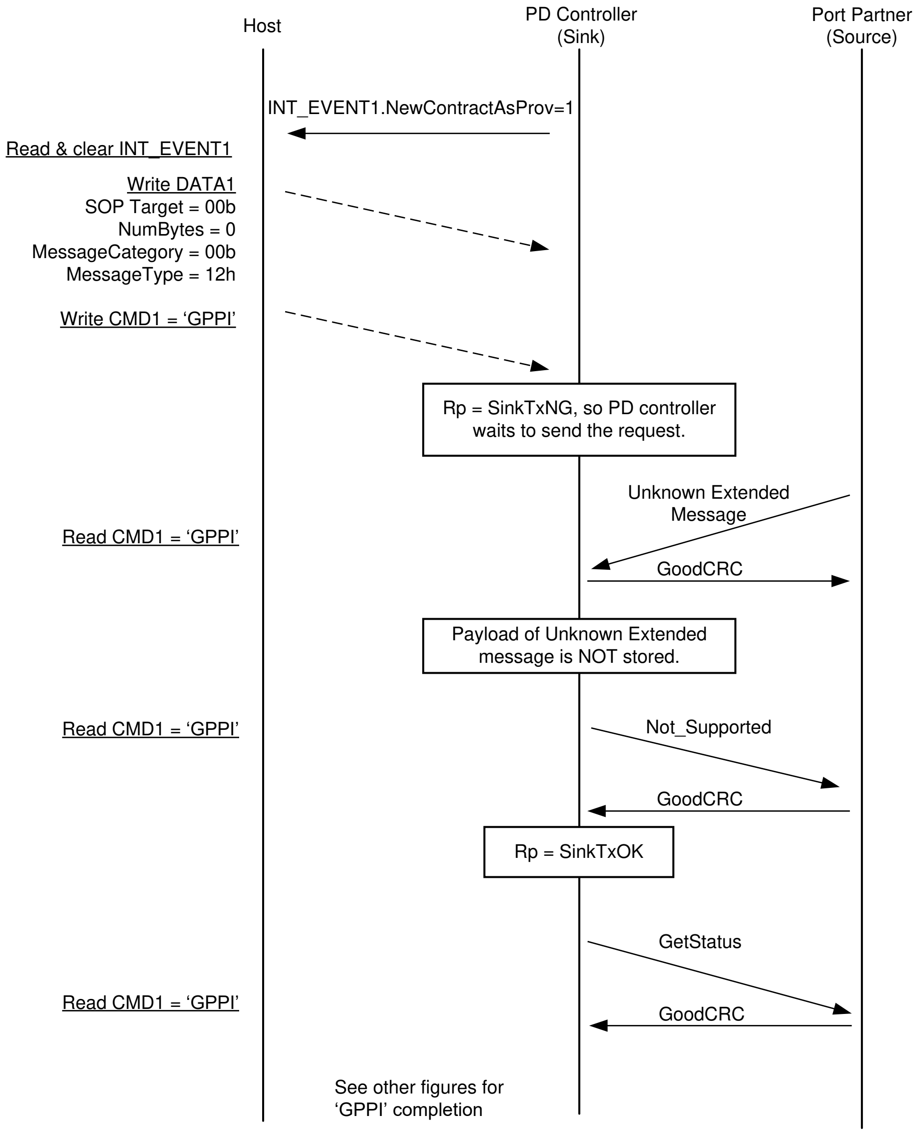


Figure 4-3. 'GPPI' interrupted by an unknown message.

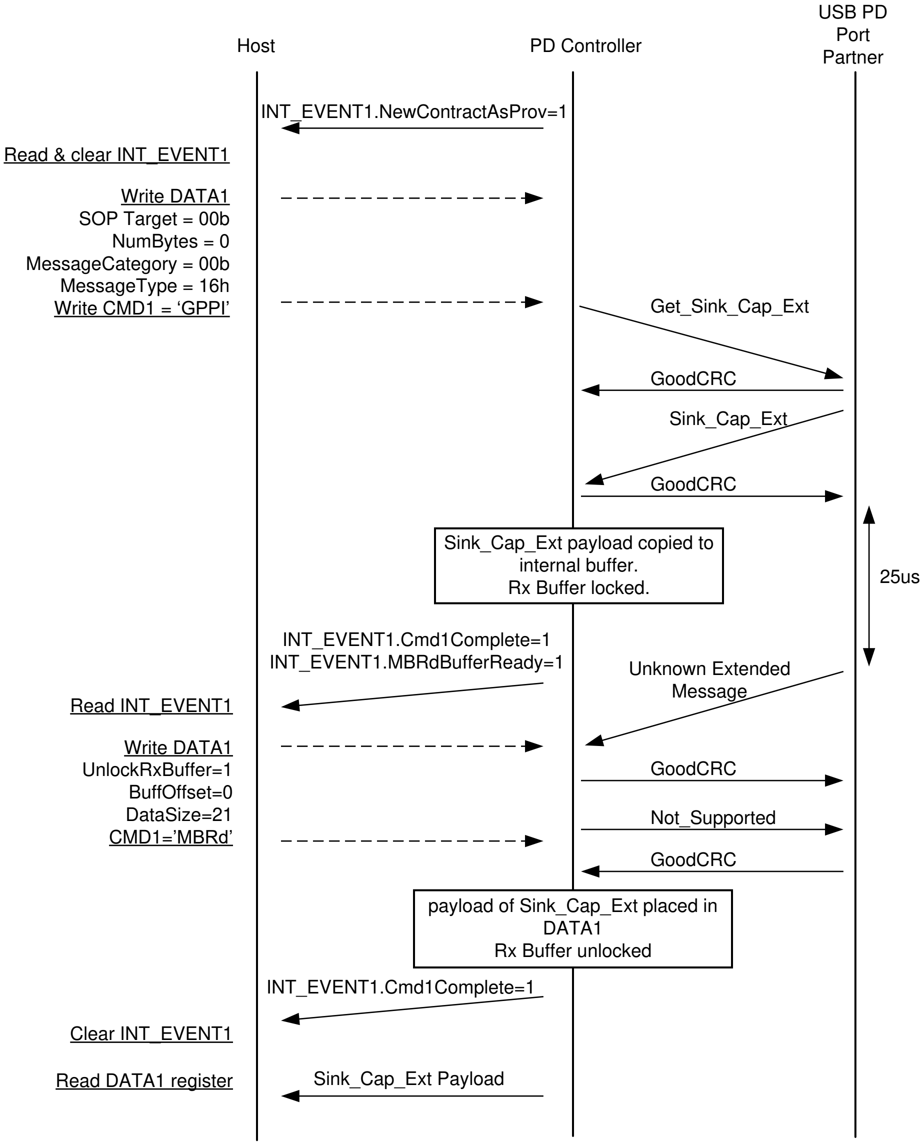


Figure 4-4. 'GPPI' interrupted by an unknown extended message.

4.4.8 'SSrC' - PD Send Source Capabilities

Table 4-12. 'SSrC' - PD Send Source Capabilities

Description	The 'SSrC' Task instructs the PD Controller to send a SourceCapabilities message at the first opportunity while maintaining policy enginecompliance.
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The 'SSrC' Task completes either when the Sink Capabilities message GoodCRC is received or the Task otherwise fails. The 'SSrC' Task shall be considered rejected if: • PD Controller is not in a Source role. The 'SSrC' Task shall be considered timed-out if: • The Source Capabilities message was sent but no GoodCRC was received. The 'SSrC' Task shall be considered successful if: • The Source Capabilities message was sent and a GoodCRC is received.
Side Effects	Other registers may change as a result of the contract negotiation that begins with the new Source Capabilities message.
Additional Information	None

4.4.9 'DRST' - PD Data Reset

Table 4-13. 'DRST' - Execute a Data Reset per USB specifications

Description	The 'DRST' Task instructs PD Controller to issue Data Reset PD message at the first opportunity while maintaining policy engine compliance.
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	This Task shall be rejected if: • The negotiated specification revision is less than 3.0 (PD3_STATUS.portNegotiatedSpecRev < 10b). This Task shall time-out if: • If the expected response from the Port Partner is not received within the time allowed by the USB PD spec. This Task shall complete successfully if: • The PD controller receives an Accept message in response to the transmitted Data Reset message.
Side Effects	Unless this Task is rejected, it will cause the PD controller to transmit a Data Reset USB PD message and begin the Data Reset process per the USB PD requirements.
Additional Information	The Data Reset process may lead to Error Recovery and reset of the connection if the Port Partner does not respond as expected.

4.4.10 'MBRd' - Message Buffer Read

Table 4-14. 'MBRd' - Read from PD message buffer.

Description	**The MBRd Task instructs the PD Controller to read data from the extended message buffer previously received from the Port Partner.**			
INPUT DATAX	**Bit**	**Name**	**Description**	
	23	Reserved	Reserved (Write as 0).	
	22	UnlockRxBuffer	This input controls whether or not the PD controller unlocks its internal buffer after this Task is completed. It is recommended to unlock the internal buffer as soon as possible to make room for other incoming messages. It is important that the host only set this bit to 1 after it has received an alert that the Rx Buffer is locked (i.e. INT_EVENTx.MBRdBufferReady asserted).	
			0b	Do not clear the internal buffer, another 'MBRd' Task may be used later.
			1b	Clear the internal buffer after this Task completes and the requested data is in the DATAx register.
	21:16	DataSize	Number of data bytes to be read in from the message buffer. Up to 62 bytes can be read at once.	
	15:0	BuffOffset	Buffer Offset. Values 0 to 259 are possible.	
OUTPUT DATAX	**Bit**	**Name**	**Description**	
	511:16	DataByte1	First Byte of data read at BuffOffset.	
	15:0	MessageSize	Size of message in bytes.	
Task Completion	The MBRd Task completes once buffer data of DataSize at BuffOffset has been read from the message buffer.			
Side Effects	None			
Additional Information	This Task is required for the host to obtain the information from the response due to the usage of the 'GPPI' Task. The PD controller has a single buffer per port that is shared for these messages.			

4.4.11 'ALRT' - Send Alert Message

Table 4-15. 'ALRT' - Send a USB PD Alert message.

Description	**The ALRT Task instructs the PD Controller to issue a Alert message to the Port Partner at the first opportunity while maintaining policy engine compliance. Contents of the Alert message sent come from** *TX_ADO* **register (0x75).**
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The Task will complete when the Alert message is sent and GoodCRC'ed or the Task is otherwise rejected. The 'ALRT' Task shall be considered rejected if: • A Not_Supported message is received in response to the Alert message. The 'ALRT' Task shall be considered timed-out if: • The Port Partner fails to respond within the time required by the PD specification. The 'ALRT' Task shall be considered successful if: • The Alert message is sent and GoodCRC'ed.
Side Effects	None
Additional Information	None

4.5 Alternate Mode Tasks

4.5.1 *'AMEn' - PD send Enter Mode*

Table 4-16. 'AMEn' - PD send Enter Mode

Description	**The PD controller will automatically send the Enter Mode Command for appropriate Alternate Modes discovered based in its configuration settings. For the normal PD state machine process of entering Alternate Modes, the Host does not need to interact. However, if the Host ever needs to manually exit an Alternate Mode, and then at a later time wants to re-enter an Alternate Mode, then this 4CC Task must be used. If the Alternate Mode trying to be re-entered requires any cable negotiation (SOP', SOP") in addition to negotiation with the other USB-C port (SOP), then the 'AMEn' Task will automatically negotiate the Alternate Mode with all the required SOP* targets.**		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 2-3: AMEn SVID		
	15:0	SVIDTarget	SVID to use for Enter Mode SVDM Task.
	Byte 1: AMEn Task Header		
	7:5	ObjPos	Object Position of Mode to enter. A value of 111b will result in the Task being rejected. Any other value will cause the PD Controller to send an Enter Mode SVDM command to the requested SOPTarget. The host should not use the value of 000b.
	4:0	Reserved	Reserved (write 0).
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	The 'AMEn' Task completes when the Enter Mode SVDM command is delivered and a GoodCRC is received and an ACK Response is received, or the Task is otherwise rejected or times-out. If PD policy does not currently allow an SVDM to be sent (for example no Explicit Contract has been achieved) this Task will wait until PD policy allows the SVDM to be sent. The PD spec currently does not allow a BUSY response to an Enter Mode SVDM command, however if PD Controller receives a BUSY Response the 'AMEn' Task will remain active and continue to retry the Enter Mode SVDM command.		
Side Effects	Assuming the 'AMEn' Task succeeds in sending the Enter Mode SVDM command and receives a SVDM Response, PD Controller will update the *STATUS* register (0x1A) to indicate that a Mode is active. For SVIDs and Modes that PD Controller supports it will also update the appropriate VID/SID Status and *Data Status* registers (for example the DPModeActive field in *DP SID Status* when a SVDM ACK is received to the DP SID Enter Mode Command for the DisplayPort Object Position). As each SVDM Response is received it will be stored in the *RX_OTHER_VDM* register (0x4F) as usual, allowing the Host to perform any additional processing of the SVDM Response. Note that if ObjPos is set to 000b Enter Mode SVDM commands may be sent to multiple SOP* Ordered Sets, meaning multiple Enter Mode SVDM Responses may be received, only the final response (most likely SOP since cable mode entry is generally done first) will remain in the *RX VDM* register.The 'AMEn' Task may be sent to a SVID and Mode that PD Controller does not recognize. In this case PD Controller will update the ModeEntered field in the *Status* register (assuming the Enter Mode was ACKed) and store the SVDM Response in the *RX VDM* register, but no other register updates will be made since PD Controller does not have any register fields that pertain to the requested SVID / Mode.		
Additional Information	None		

4.5.2 'AMEx' - PD send Exit Mode

Table 4-17. 'AMEx' - PD send Exit Mode

Description	**The PD Controller will automatically discover and enter into Alternate Modes according to its configuration settings. It will also exit Alternate Modes on its own whenever needed to maintain PD compliance. However, the Host may command the PD controller to exit the Alternate Mode using this Task. Once an Alternate Mode has been exited using 'AMEx', then the PD controller will only enter the Alternate Mode if commanded to via 'AMEn'. When using the 'AMEx" Task to exit an Alternate Mode, all SOP* targets that require an Exit Mode SVDM command to properly exit the Alternate Mode will be sent an Exit Mode SVDM command.**		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 2-3: AMEx SVID		
	15:0	SVIDTarget	SVID to use for Exit Mode SVDM command.
	Byte 1: AMEx Task Header		
	7:5	ObjPos	Object Position of Mode to exit. If 111b PD Controller will use 111b in its Exit Mode SVDM command to exit all Modes associated with the SVID. SOPTarget should be 11b when this option is used. 000b is Reserved and the 'AMEx' Task will be rejected if used.
	4:0	Reserved	Reserved (write 0).
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	The 'AMEx' Task completes when the Exit Mode SVDM command(s) are delivered and a GoodCRC is received and an ACK Response is received, or the Task is otherwise rejected or times-out. There should be no reason for PD policy to prevent the SVDM to be sent or else a Mode should not have been able to be entered in the first place. The PD spec currently does not allow a BUSY response to an Exit Mode SVDM command, however if PD Controller receives a BUSY Response the 'AMEx' Task will remain active and continue to retry the Exit Mode SVDM command.		
Side Effects	Assuming the 'AMEx' Task succeeds in sending the Exit Mode SVDM command(s) and receives a SVDM Response PD Controller will take the appropriate action for exiting the Mode. The PD Controller will also update the appropriate VID/SID Status and *Data Status* registers (for example the DPModeActive field in *DP SID Status* when a SVDM ACK is received to the DP SID Exit Mode Command for the DisplayPort Object Position). As each SVDM Response is received it will be stored in the *RX_OTHER_VDM* register (0x4F) as usual, allowing the Host to perform any additional processing of the SVDM Response. Note that because the PD controller will send Exit Modes messages on all SOP targets as appropriate, the SOP' and SOP" Exit Mode SVDM Responses may be overwritten by additional Exit Mode SVDM Responses, since the last Exit Mode is sent to SOP, its SVDM Response will be the final value in the *RX VDM* register.		
Additional Information	None		

4.5.3 'AMDs' - PD Start Alternate Mode Discovery

Table 4-18. 'AMDs' - Start discovery process

Description	The 'AMDs' Task instructs PD Controller to start the Alternate ModeDiscovery process.
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	This task will complete after the PD controller has sent the Discover Identity, Discover SVIDs, and Discover Modes messages that take place during that Alternate Mode Discovery process, or when the task is otherwise rejected. The 'AMDs' Task shall be considered rejected if: • Not a DFP for PD2.0 operation. • The Discover Identity message was sent and GoodCRC'ed and a NAK response was received. • The Discover SVIDs message was sent and GoodCRC'ed and a NAK response was received. The 'AMDs' Task shall be considered successful if: • Both the Discover Identity and the Discover SVIDs messages were sent and GoodCRC'ed, and ACK Responses were returned for both of them.
Side Effects	As a part of the Alternate Mode Discovery process, the DISCOVERED_SVIDS register (0x21), RX_IDENTITY_SOP register (0x48), RX_IDENTITY_SOPp register (0x49), USER_VID_STATUS register (0x57), DP_SID_STATUS register (0x58), CUSTOMD_STATUS (0x5A) and INTEL_VID_STATUS register (0x59) could be updated. Which registers get updated will depend on how many of the Alternate Mode Discovery process PD message successfully get responses, which also influences whether or not the Task returns it completed successfully or not in the Output DATAx register
Additional Information	None

4.5.4 'GCdm' - Get Custom Discovered Modes

Table 4-19. 'GCdm' - Get custom discovered modes

Description	**After a successful 'GCdm' Task, PD Controller returns a list of VDOs along with their respective object position.**		
INPUT DATAX	**Bit**	**Name**	**Description**
	7:0	Reserved	Reserved
	23:8	SVID	SVID
OUTPUT DATAX	**Bit**	**Name**	**Description**
	31:0	VDOMode1	VDO for Mode 1
	39:32	VDOMode1Pos	Object Position for Mode 1
	71:40	VDOMode2	VDO for Mode 2
	79:72	VDOMode2Pos	Object Position for Mode 2
	111:80	VDOMode3	VDO for Mode 3
	119:112	VDOMode3Pos	Object Position for Mode 3
	151:120	VDOMode4	VDO for Mode 4
	159:152	VDOMode4Pos	Object Position for Mode 4
	191:160	VDOMode5	VDO for Mode 5
	199:192	VDOMode5Pos	Object Position for Mode 5
	231:200	VDOMode6	VDO for Mode 6
	239:232	VDOMode6Pos	Object Position for Mode 6
	271:240	VDOMode7	VDO for Mode 7
	279:272	VDOMode7Pos	Object Position for Mode 7
Task Completion	The 'GCdm' Task completes when Output is updated with VDO modes and positions.		
Side Effects	None		
Additional Information	This 'GCdm' Task does not initiate sending of any USB PD messages. Instead, it just returns information gathered during the discovery phase automatically done by the PD controller. Therefore, it should not be issued until INT_EVENTx.DiscoverModeComplete has been asserted. The PD controller does not generally discover modes for a SVID that is disabled, so this 'GCdm' task should only be issued with a SVID that has been enabled such as USER_VID_CONFIG.User_AltMode_SVID_Value when USER_VID_CONFIG.UserVidEnabled is asserted.		

4.5.5 'VDMs' - PD send VDM

Table 4-20. 'VDMs' - PD send VDM

Description	The 'VDMs' Task instructs PD Controller to send a Vendor DefinedMessage (VDM) at the first opportunity while maintaining policy enginecompliance.			
INPUT DATAX	Bit	Name	Description	
	Byte 31: Initiator Wait State Timer Configuration			
	7:0	InitiatorWaitTimer	Configurable Initiator Wait State Timer. Please note, this timer is only used if the InitiatorResponder bit in byte 30 is set to 1b. The PD controller will wait for this amount of time for a response. (1ms per LSB)	
	Byte 30: VDMs Configuration			
	7:1	Reserved		
	0	InitiatorResponder	The VDMs may be sending a response or initiating a sequence.	
			0b	This is a response so the PD controller will transmit the message regardless of the collision avoidance Rp value. Example, an ACK message is responding.
			1b	This is initiating a VDM sequence, so the PD controller will follow USB PD collision avoidance requirements. Example, a REQ message is initiating a new VDM sequence.
	Bytes 26-29: VDO #7 (treated as 32-bit little endian value)			
	31:0	VDO7	Contents of seventh VDO, if applicable.	
	Bytes 22-25: VDO #6 (treated as 32-bit little endian value)			
	31:0	VDO6	Contents of sixth VDO, if applicable.	
	Bytes 18-21: VDO #5 (treated as 32-bit little endian value)			
	31:0	VDO5	Contents of fifth VDO, if applicable.	
	Bytes 14-17: VDO #4 (treated as 32-bit little endian value)			
	31:0	VDO4	Contents of fourth VDO, if applicable.	
	Bytes 10-13: VDO #3 (treated as 32-bit little endian value)			
	31:0	VDO3	Contents of third VDO, if applicable.	
	Bytes 6-9: VDO #2 (treated as 32-bit little endian value)			
	31:0	VDO2	Contents of second VDO, if applicable.	
	Bytes 2-5: VDO #1 (treated as 32-bit little endian value)			
	31:0	VDO1	Contents of first VDO (VDM Header if SVDM).	
	Byte 1: VDMs Task Header			
	7	AMIntrusiveModeResponse	When set this message satisfies a pending AMIntrusiveMode interaction, PD Controller will stop sending BUSY Responses to the last received SVDM command.	
	6	Reserved	write as 0b	
	5:4	SOPTarget	Ordered Set to send VDM to.	
			00b	SOP.
			01b	SOP'.
			10b	SOP''.
			11b	SOP*_Debug (SOP'_Debug for Source, SOP''_Debug for Sink).
	3	Version	To maintain backwards compatibility, this field is used to indicate whether bytes 30-31 are ignored or used.	
			0b	VDMs version 1 (ignores bytes 30-31). The PD controller always waits 30ms for a response.
			1b	VDMs version 2 (implements bytes 30-31)
	2:0	NumDOs	Number of VDOs to transmit (1-7), includes VDM Header for SVDMs.	
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.			

Table 4-20. 'VDMs' - PD send VDM (continued)

Description	The 'VDMs' Task instructs PD Controller to send a Vendor DefinedMessage (VDM) at the first opportunity while maintaining policy enginecompliance.
Task Completion	The 'VDMs' Task completes when the VDMs is delivered and a GoodCRC is received or the appropriate number of retries have been attempted without a GoodCRC, or the Task is rejected. This Task does not wait for a VDM response since there is no guarantee of a response especially for Unstructured VDMs. The 'VDMs' Task shall be considered rejected if: • NumDOs is set to 0. • PD policy does not allow a VDM to be sent at this time. • DFP/UFP is instructed to send to SOP' when not appropriate (e.g. DFP during Implicit Contract following a PR_Swap) or a UFP is instructed to send to SOP''. The 'VDMs' Task shall be considered timed-out if: • The VDM was sent but no GoodCRC was received. The 'VDMs' Task shall be considered successful if: • The VDM was sent and a GoodCRC was received.
Side Effects	If the 'VDMs' Task succeeds in sending the requested VDM, PD Controller is not aware of the VDM it sent, so it will not be expecting a response. All incoming VDMs that are not Initiator Attention messages will be stored in the *RX_OTHER_VDM* register (0x4F) regardless of PD Controller 's current state so the response can be processed by the Host, but it will not otherwise be processed by PD Controller . For example, if VDMs is used to send a Discover Identity SVDM Command to SOP', the *RX Identity SOP'* register does not get updated since PD Controller 's PD state machine was not in the proper state to receive this response. There may be some delay before the VDM is transmitted. See the table below for details.
Additional Information	The host should not send back-to-back 'VDMs' command to quickly. In particular, if the SOPTarget was not SOP, then the host shall wait for at least 35 ms while the PD controller awaits a response from the cable. It is recommended that the host wait 35 ms in between consecutive 'VDMs' messages for SOPTarget=SOP as well.

The table below summarizes the situations for which the PD controller will delay transmission of the VDM.

Table 4-21. Description of delay before sending the VDM for 'VDMs'.

PD_STATUS	PD3_STATUS	VDO1[15]	VDO1[7]	VDO1[6]	Description of Delay
PresentPDRole	portNegotiatedSpecRev				
X	01b (PD2)	X	X	X	No Delay
0b (Sink)	10b (PD3)	X	X	X	Delayed until PD_STATUS.CCPullUp=11b (SinkTxOK).
1b (Source)		0b[1]	X	X	The PD controller will act as the Atomic Message Sequence Initiator. USB PD collision avoidance requirements may cause a delay of tSinkTx before the VDM is sent.
		1b[2]	0b	0b	
			0b	1b	No delay
			1b	0b	
			1b	1b	

[1] Unstructured VDM.
[2] Structured VDM (SVDM)

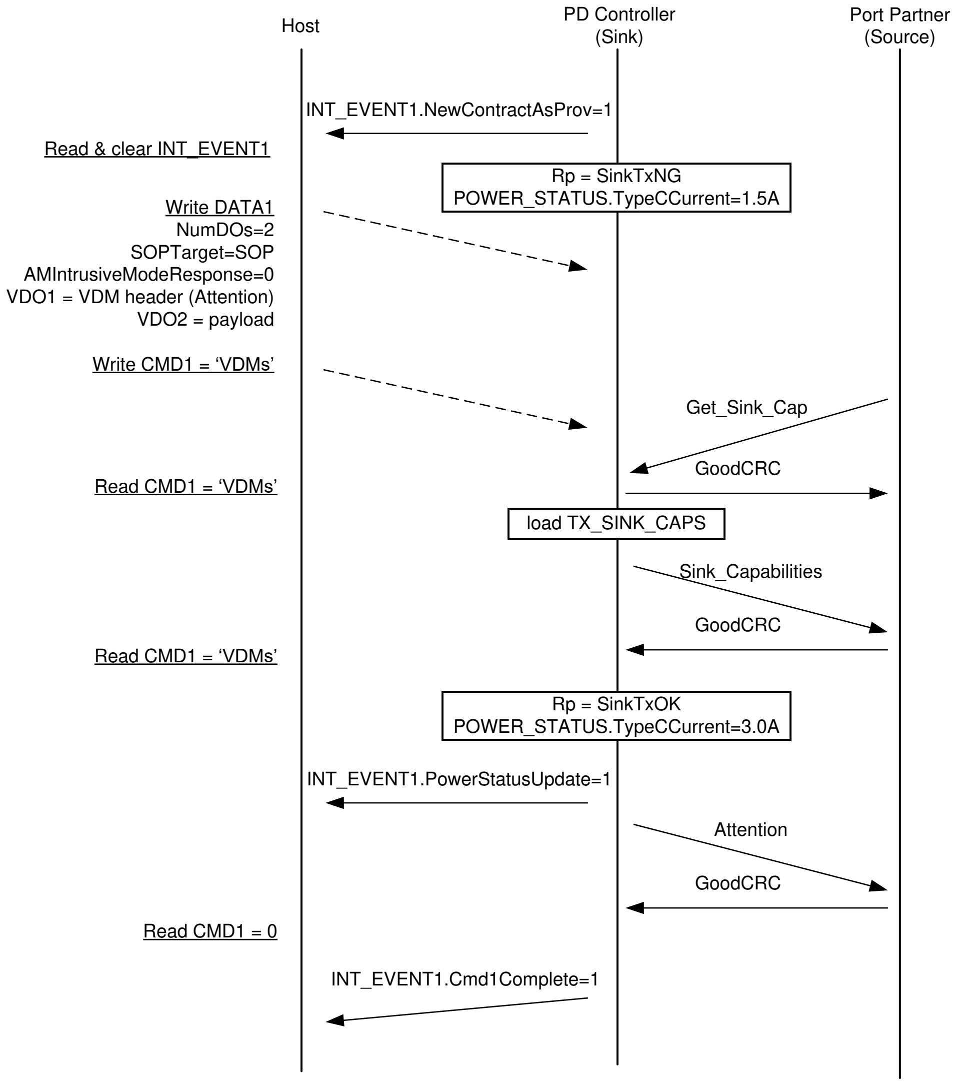


Figure 4-5. 'VDMs' example sequence.

4.6 Power Switch Tasks

4.6.1 'SRDY' - System ready to sink power

Table 4-22. 'SRDY' - System ready to sink power

Description	**The 'SRDY' Task instructs PD Controller to enable a power switch forinput.**			
INPUT DATAX	**Bit**	**Name**	**Description**	
	Byte 1: SRDY Task Header			
	7:3	Reserved	Reserved (write 0).	
	2:0	SwitchSelect	Specifies which switch will be enabled. Switches must be configured as inputs that wait for SYS_RDY Task (SRDY).	
			000b	PP1 (PP_5V1)
			001b	PP2 (PP_5V2)
			010b	PP3 (PP_EXT1)
			011b	PP4 (PP_EXT2)
			100b - 101b	Reserved
			110b	Automatically-selected by the PP*Config field in the GLOBAL_SYSTEM_CONFIG register (0x27) . Assumes a single switch is configured as an input. This allows a Host to issue this Task without having to know which switch that is.
			111b	Automatically-selected by PD Controller policy. Used primarily to re-enable a switch that has been turned off for some reason.
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.			
Task Completion	The 'SRDY' Task completes when the selected input switch is successfully enabled or the Task otherwise fails.			
Side Effects	When 'SRDY' completes power switches may have been re-configured, which will affect the Status register.			
Additional Information	When this 'SRDY' Task is issued, the selected switch will enable with soft-start. The turn-on time of the switch and the system load should be considered along with the Safe-Operating Area (SOA) of the switch. In most cases, the system should draw a very small load until the switch is enabled.			

4.6.2 'SRYR' - SRDY reset

Table 4-23. 'SRYR' - SRDY reset

Description	The 'SRYR' Task instructs PD Controller to disable the currently-enabled input switch, if there is one. This command applies only to the port to which it is addressed (via I2C slave address). The sink port for the other port (for dual-port controllers) is not affected.
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The 'PBMs' Task completes once output has a valid PatchStartStatus. If MODE register (0x03) is equal to 'APP ', then this Task will be rejected.
Side Effects	When the 'PBMs' is successful, the second slave address will be set to the input value.
Additional Information	The host can only issue a 'PBMs' Task to the I2C_EC port of the PD controller. If the host issues 'PMBs' a second time, then the PD controller ignores the DATAX input, restarts the burst-mode timer, and resets the pointer to the beginning of the patch space in RAM. If the MODE register is 'APP ' indicating that the PD controller is in the APP mode, then it will reject the 'PBMs' Task.

4.7 Patch Bundle Update Tasks

The following tasks are used for updating a Patch Bundle.

4.7.1 'PBMs' - Start Patch Burst Mode Download Sequence

Table 4-24. 'PBMs' - Start Patch Burst Download Sequence

Description	**The 'PBMs' Task starts the patch loading sequence. This Task initializes the firmware in preparation for a patch bundle load sequence and indicates what the patch bundle will contain.**			
INPUT DATAX	**Bit**	**Name**	**Description**	
	Byte 6: Burst Mode Timeout			
	7:6	Reserved		
	5:0	Timeout value	Timeout value for this task. A non-zero value must be used, it is recommended to always use 0x32 in this field (5 seconds) (LSB of 100ms).	
	Byte 5: I2C slave for downloading patch.			
	7	Reserved		
	6:0	I2C Slave Address	The following slave addresses are not valid: • 0x00 • The I2C1s slave address of any port selected using the ADCINx pins. Refer to data-sheet.	
	Bytes 0-3: Low Region Binary bundle size in of bytes: [Byte4, Byte3, Byte2, Byte1]			
	39:32	Byte4 of bundle size		
	31:24	Byte3 of bundle size		
	23:16	Byte2 of bundle size		
	15:8	Byte1 of bundle size		
OUTPUT DATAX	**Bit**	**Name**	**Description**	
	7:0	PatchStartStatus	Status of the patch start.	
			0x00	Patch start success
			0x04	Invalid bundle size
			0x05	Invalid slave address
			0x06	Invalid Timeout value
Task Completion	The 'PBMs' Task completes once output has a valid PatchStartStatus. If MODE register (0x03) is equal to 'APP ', then this Task will be rejected.			
Side Effects	When the 'PBMs' is successful, the second slave address will be set to the input value.			
Additional Information	The host can only issue a 'PBMs' Task to the I2C_EC port of the PD controller. If the host issues 'PMBs' a second time, then the PD controller ignores the DATAX input, restarts the burst-mode timer, and resets the pointer to the beginning of the patch space in RAM. If the MODE register is 'APP ' indicating that the PD controller is in the APP mode, then it will reject the 'PBMs' Task.			

4.7.2 'PBMc' - Patch Burst Mode Download Complete

Table 4-25. 'PBMc' - Patch Burst Download Complete

Description	**The 'PBMc' Task ends the patch loading sequence. Send this Task after all patch data has been transferred. This Task will initiate the CRC check on the binary patch data that has been transferred, and if the CRC is successful, the patch_init function contained within the patch will be executed.**			
INPUT DATAX	None			
OUTPUT DATAX	**Bit**	**Name**	**Description**	
	319:288	acCalculatedCRC	The CRC calculated in FW for the configuration data.	
	287:256	acTransferredCRC	The CRC transferred along with the configuration data	
	255:240	Reserved	reads as 0	
	239:224	acIndicatedDataSize	The indicated DataSize in the transferred configuration data.	
	223:216	acHeaderVersion	The indicated header version in the transferred configuration data.	
	215:208	acFailCode	An error code indicating why the app config data failed to apply, if it failed to apply	
			0x00	AC_FAIL_NONE: No failure
			0x01	AC_FAIL_WRONG_HEADER_VERSION: The header version is expected to be 1 and was not
			0x02	AC_FAIL_TOO_MUCH_DATA: The DataSize field indicates that you are trying to load more configuration data that there is allocated SRAM for
			0x03	AC_FAIL_CRC_CHECK_FAIL: The CRC comparison failed
	207:200	acState	The current internal state of the AppConfig state machine	
			0x00	AC_NODATA: No configuration data found yet, because we haven't started looking
			0x01	AC_LOADING_DEFAULT: Attempting to load configuration data from a factory default
			0x02	AC_LOADING_SRAM: Attempting to load configuration data from SRAM
			0x03	AC_LOADING_FLASH: Attempting to load configuration data from Flash
			0x04	AC_LOADING_I2C: Attempting to load configuration data from I2C
			0x05	AC_LOADING_DONE: Done loading configuration data, we found valid data
			0x06	AC_ERROR: A generic error state
			0x07	AC_DONE_SUCCESS: Completely done with the app customization process and the records were applied successfully.
			0x08	AC_DONE_FAIL: Completely done with the app customization process and the records were not applied
	199:192	configBundleGood	1 if the top-level state machine found a valid configuration bundle, otherwise 0.	
	191:160	rpRomVersionExpected	The romVersionExpected in the transferred bundle's patch header	
	159:144	rpBundleTotalSize	The bundleTotalSize in the transferred bundle's patch header	
	143:128	rpBundleFlags	The bundleFlags in the transferred bundle's patch header	
	127:96	rpPatchBodyCrc	The patchBodyCrc in the transferred bundle's patch header	
	95:64	rpPatchHeaderCrc	The patchHeaderCrc in the transferred bundle's patch header	

Table 4-25. 'PBMc' - Patch Burst Download Complete (continued)

Description	**The 'PBMc' Task ends the patch loading sequence. Send this Task after all patch data has been transferred. This Task will initiate the CRC check on the binary patch data that has been transferred, and if the CRC is successful, the patch_init function contained within the patch will be executed.**			
OUTPUT DATAX	**Bit**	**Name**	**Description**	
	55:48	rpBundleSignature	The bundleSignature in the transferred bundle's patch header	
	47:40	rpState	The current internal state of the RomPatch state machine.	
			0x00	RP_NOPATCH: No patch has been loaded
			0x01	RP_LOADING: In the process of loading patch data
			0x02	RP_LOADINGDONE: All patch data has been received
			0x03	RP_RUNNING: A patch has been loaded and is running. Could also indicate that a NULL patch is active.
			0x04	RP_EARLYLOAD_SKIPPED: Indicates that the early boot process does not need to wait for a patch over I2C
			0x05	RP_UARTBOOTED: Checking for a patch in RAM
			0x06	RP_ERROR: A generic error state
	39:32	patchBundleGood	0x01 if the top-level state machine found a good ROM patch, otherwise 0x00.	
	31:24	AppConfigPatchCompleteStatus	0x00	
			0x40	Warning
			0x80	Failure
	23:16	DevicePatchCompleteStatus	A return code indicating whether the RomPatch state machine executed successfully. This value is always valid, and reflective of the internal state of the RomPatch mechanism, but should only be considered if the bundle transferred did in fact include patch data.	
			0x00	Success
			0x20	Not ready
			0x40	Not a patch
			0x41	Patch header checksum mismatch
			0x42	Patch not compatible with this version of ROM
			0x43	Patch code checksum mismatch
			0x44	Null patch received
			0x45	Error patch received
	15:8	cpReturn	Always returns success, there is no way for it to fail.	
	Byte 1: Return Code			
	7:4	rpReturnIndicator	The most significant nibble of the rpReturn value.	
			0x0	Success
			0x2	Informational
			0x4	Warning
			0x8	Error
	3:0	acReturnIndicator	The most significant nibble of the acReturn value.	
			0x0	Success
			0x2	Informational
			0x4	Warning
			0x8	Error
Task Completion	The 'PBMc' Task completes as output has a valid DevicePatchCompleteStatus and AppConfigPatchCompleteStatus. This Task is rejected if the DATAX input does not contain the total patch size. If MODE register (0x03) is equal to 'APP ', then this Task will be rejected.			
Side Effects	Before this Task completes it will change the I2C slave address from the patch address back to the normal value. Upon successful completion of this Task the PD controller will change the MODE register (0x03) to 'APP ' and move to the application mode.			
Additional Information	When the CMDx register goes to 0 check the Output DATAX register for status. If the MODE register is 'APP ' indicating that the PD controller is in the APP mode, then it will reject the 'PBMc' Task.			

4.7.3 'PBMe' - End Patch Burst Mode Download Sequence

Table 4-26. 'PBMe' - Patch Burst Mode Exit

Description	The 'PBMe' Task ends the patch loading sequence. This Task instructs the PD controller to complete the patch loading process.
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The 'PBMe' Task completes once it has ended the patch loading sequence. If MODE register (0x03) is equal to 'APP ', then this Task will be rejected.
Side Effects	When the 'PBMe' is successful, the second slave address will be restored to the value configured by the ADCINx pins. The PD controller leaves the MODE register (0x03) as 'PTCH' and will wait for the patching process to restart.
Additional Information	If the MODE register is 'APP ' indicating that the PD controller is in the APP mode, then it will reject the 'PBMe' Task.

4.7.4 'GO2P' - Go to Patch Mode

Table 4-27. 'GO2P' - Forces PD controller to return to 'PTCH' mode and wait for patch over I2C.

Description	The 'GO2P' Task causes the PD controller to re-enter the patch mode (MODE = 'PTCH').
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The 'GO2P' Task completes once the PD controller has re-entered the patch mode. • If the PD controller has re-entered the patch mode and the MODE register reads as 'PTCH'. The 'GO2P' Task is considered rejected if: • The PD controller did not enter the 'APP ' mode without receiving a patch over I2C. • BOOT_STATUS.PatchConfigSource does not read as 3h or 4h.
Side Effects	When the 'GO2P' Task is successful, the MODE register will read as 'PTCH' and the USB PD PHY will be disabled. The PD Controller may temporarily NAK I2C transactions. The host should wait for the IRQ signal to assert (because INT_EVENT1.ReadyForPatch is asserted), and then push the patch as soon as possible.
Additional Information	The 'GO2P' Task should only be used when the ADCINx configuration option NegotiateHighVoltage is used.

4.7.5 'FLrd' - Flash Memory Read

Table 4-28. 'FLrd' - External EEPROM Read

Description	**The 'FLrd' Task reads the flash at the specified address.**		
INPUT DATAX	**Bit**	**Name**	**Description**
	31:0	Flash Address	Flash Address
OUTPUT DATAX	**Bit**	**Name**	**Description**
	127:0	Memory Contents	Memory contents (little-endian).
Task Completion	The 'FLrd' Task completes once selected memory locations are loaded.		
Side Effects	The PD controller ignores the I2C3m_IRQ pin until this Task is completed.		
Additional Information	None		

4.7.6 'FLad' - Flash Memory Write Start Address

Table 4-29. 'FLad' - External EEPROM Start Address

Description	**The 'FLad' Task sets start address in preparation the flash write.**		
INPUT DATAX	**Bit**	**Name**	**Description**
	31:0	Flash Address	Flash address (treated as 32-bit little-endian value).
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	The 'FLad' Task completes once selected memory address is loaded.		
Side Effects	The PD controller ignores the I2C3m_IRQ pin until this Task is completed.		
Additional Information	None		

4.7.7 'FLwd' - Flash Memory Write

Table 4-30. 'FLwd' - External EEPROM Memory Write

Description	**The 'FLwd' Task writes data beginning at the flash start address defined by the 'FLad' Task. The address is auto-incremented.**			
INPUT DATAX	**Bit**	**Name**	**Description**	
	511:0	Flash Address	Up to 32 bytes of flash data.	
OUTPUT DATAX	**Bit**	**Name**	**Description**	
	7:0	ReturnCode	Status of write.	
			0x00h	Flash memory write successful
			0xFFh	Error, flash is busy
Task Completion	The 'FLwd' Task completes once selected the flash is written.			
Side Effects	The PD controller ignores the I2C3m_IRQ pin until this Task is completed.			
Additional Information	None			

4.7.8 'FLvy' - Flash Memory Verify

Table 4-31. 'FLvy' - External EEPROM Verify

Description	**The 'FLvy' Task verifies if the patch/configuration is valid.**			
INPUT DATAX	**Bit**	**Name**	**Description**	
	31:0	Flash Address	Flash Address	
OUTPUT DATAX	**Bit**	**Name**	**Description**	
	7:0	ReturnCode		
			0x00h	The patch/configuration is valid.
			0x01h	The patch/configuration is not valid.
Task Completion	The 'FLvy' Task completes once header is checked and validated.			
Side Effects	The PD controller ignores the I2C3m_IRQ pin until this Task is completed.			
Additional Information	None			

4.8 System Tasks

4.8.1 'ABRT' - Abort current Task

Table 4-32. 'ABRT' - Abort current task

Description	**The 'ABRT' Request is not exactly a Task of its own, it is a value that when written to a CMDX register can affect a long-running Task that is currently running on that CMDX interface. Normally a Host is not allowed to write anything to CMDX if it reads back as anything other than 0 or '!CMD', but the 'ABRT' value can always be written to CMDX.**
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	As the 'ABRT' Request is not actually a task it is difficult to say when it completes. If no Task was active the top-level command processing loop will simply ignore it and clear CMDX. If a Task is running then 'ABRT' will remain in CMDX until the Task completes (either because it completed normally or because it detected the 'ABRT' and aborted).
Side Effects	Writing 'ABRT' to a CMDX register should have no side effects other than canceling the active Task. The aborted Task should perform any necessary cleanup.
Additional Information	None

4.8.2 'ANeg' - Auto Negotiate Sink Update

Table 4-33. 'ANeg' - Re-evaluate the auto-negotiate sink register

Description	**The 'ANeg' Task instructs PD Controller to re-evaluate the** *Auto Negotiate Sink* **register (0x37). If the re-evaluation produces a different RDO than the Active Contract RDO then a new Request message is sent.**
INPUT DATAX	None
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.
Task Completion	The 'ANeg' Task completes once the new RDO is calculated and PD Controller either decides to send a new Request message (and that message is sent and the GoodCRC received) or determines that no Request is necessary. The task is rejected of the PD controller is not in a sink power role.
Side Effects	The side effects include a new PD contract negotiation and updates to the associated registers.
Additional Information	None

4.8.3 'DBfg' - Clear Dead Battery Flag

Table 4-34. 'DBfg' - Clear Dead Battery Flag

Description	The 'DBfg' Task is used to clear the dead battery flag. This Task does not disable the PP_EXT input switch that may have been enabled during dead battery operation.
INPUT DATAX	None
OUTPUT DATAX	None
Task Completion	The 'DBfg' Task completes once the effects of clearing the Dead Battery Flag are complete.
Side Effects	The Dead Battery Flag causes the PD Controller to take specific actions, so clearing this flag will have side effects. PD Controller 's power input is forced to VBUS until the Dead Battery Flag is cleared, so executing this Task will change PD Controller 's power input.
Additional Information	None

There are several limitations placed on the PD controller while the Dead-Battery Flag is asserted (PowerPathStatus.PowerSource = 10b).

- Fast-Role swap is not supported (on either port).
- A Hard Reset will not be transmitted while in the sink role (on either port).
- VBUS is selected as the main supply for the PD controller, even if the 3.3 V input is present.
- The PD controller will reject PR_Swap requests to become source (on either port).
- The 2nd port in the PD controller that is unconnected will only offer the USB Type-C Default Rp (PortControl.TypeCCurrent is ignored) if it connects as a source.
- A port connected to a source will only act as a Type-C sink regardless of the configuration.
- If no Source Capabilities message is received after the boot process is complete (Status.ActingAsLegacy=11b), the PD controller will not send a Hard Reset until the Dead-Battery Flag is cleared even if the SinkWaitCapTimer expires.

4.8.4 'MuxR' - Error handling for I2C3m transactions

Table 4-35. 'MuxR' - Repeats transactions on I2C3m under certain conditions.

<table>
<tr><th>Description</th><th colspan="3">'MuxR' - this task provides a way for the host to handle errors that occur on the I2C master port.</th></tr>
<tr><td rowspan="12">INPUT DATAX</td><td>Bit</td><td>Name</td><td>Description</td></tr>
<tr><td>15:10</td><td>Reserved</td><td></td></tr>
<tr><td>9</td><td>USB3RetryOnAMDRenoirMux</td><td>If this bit is asserted, the PD controller will use I2C3m to attempt to program the AMD Renoir Mux into USB3 configuration (taking into account polarity).</td></tr>
<tr><td>8</td><td>EnRetryOnSlaveAddrTbt</td><td>If this bit is asserted, the PD controller will use I2C3m to write the DATA_STATUS register with I2CMASTER_CONFIG.SlaveAddrTbt1 as the slave address. It will also repeat the write using I2CMASTER_CONFIG.SlaveAddrTbt2 as the slave address.</td></tr>
<tr><td>7</td><td>EnRetryOnSlaveAddr8</td><td>If this bit is asserted, the PD controller will retry the last transaction on the I2C3m port that used I2CMASTER_CONFIG.SlaveAddr8 as the slave address.</td></tr>
<tr><td>6</td><td>EnRetryOnSlaveAddr7</td><td>If this bit is asserted, the PD controller will retry the last transaction on the I2C3m port that used I2CMASTER_CONFIG.SlaveAddr7 as the slave address.</td></tr>
<tr><td>5</td><td>EnRetryOnSlaveAddr6</td><td>If this bit is asserted, the PD controller will retry the last transaction on the I2C3m port that used I2CMASTER_CONFIG.SlaveAddr6 as the slave address.</td></tr>
<tr><td>4</td><td>EnRetryOnSlaveAddr5</td><td>If this bit is asserted, the PD controller will retry the last transaction on the I2C3m port that used I2CMASTER_CONFIG.SlaveAddr5 as the slave address.</td></tr>
<tr><td>3</td><td>EnRetryOnSlaveAddr4</td><td>If this bit is asserted, the PD controller will retry the last transaction on the I2C3m port that used I2CMASTER_CONFIG.SlaveAddr4 as the slave address.</td></tr>
<tr><td>2</td><td>EnRetryOnSlaveAddr3</td><td>If this bit is asserted, the PD controller will retry the last transaction on the I2C3m port that used I2CMASTER_CONFIG.SlaveAddr3 as the slave address.</td></tr>
<tr><td>1</td><td>EnRetryOnSlaveAddr2</td><td>If this bit is asserted, the PD controller will retry the last transaction on the I2C3m port that used I2CMASTER_CONFIG.SlaveAddr2 as the slave address.</td></tr>
<tr><td>0</td><td>EnRetryOnSlaveAddr1</td><td>If this bit is asserted, the PD controller will retry the last transaction on the I2C3m port that used I2CMASTER_CONFIG.SlaveAddr1 as the slave address.</td></tr>
<tr><td>OUTPUT DATAX</td><td colspan="3">Byte 1: Standard Task Return Code. See also Table 4-1.</td></tr>
<tr><td>Task Completion</td><td colspan="3">If the I2C master queue is full or if the I2C master is disabled, then this task is rejected. Otherwise, the task will complete successfully.</td></tr>
<tr><td>Side Effects</td><td colspan="3">One or more I2C transactions may be repeated on the I2C master port (I2C3m).</td></tr>
<tr><td>Additional Information</td><td colspan="3">If all bits in DATAX are set to zero when CMDx is written as 'MuxR', then the PD controller clears its history of the last events for slave addresses 1 to 8. If EnRetryOnSlaveAddrX is asserted and no transaction has occurred on that I2C address since the history was cleared, then the PD controller will not execute any I2C transaction on that slave address as there is no event to repeat.
The contents of the last transaction on a given I2C slave address is defined through the App Config loaded as part of the patch bundle. The BINARYDATA_INDICES register (0x62) contains the various events that could occur on each slave address along with the payload for the transaction associated with each. When the 'MuxR' Task is executed, then for each EnRetryOnSlaveAddr* that is asserted it will be as if the last event for that slave address has occurred again.</td></tr>
</table>

4.8.5 'Trig' - Trigger an Input GPIO Event

Table 4-36. 'Trig' - Emulate a GPIO input event

Description	The 'Trig' Task may be used to trigger an input GPIO Event (see Table 3-94 for a list), The GPIO Event need not be assigned to any physical GPIO pin, so this allows implementing virtual input GPIO events.			
INPUT DATAX	Bit	Name	Description	
	15:8	GPIO Event	69d (45h)	MRESET: No assumption is made about the initial state of this virtual GPIO. No action is taken until 'Trig' is received.
			56d (38h)	I2C3_MASTER_IRQ_EVENT: No assumption is made about the initial state of this virtual GPIO. No action is taken until 'Trig' is received.
			47d (2Fh)	Prevent_High_Current_Contract_Event: This virtual GPIO is assumed to be high until 'Trig' is received. The GLOBAL_SYSTEM_CONFIG.EnableSPM bit must be asserted in order to use this virtual GPIO.
			42d (2Ah)	Retimer_SoC_OVR_Force_PWR_Event: This virtual GPIO is assumed to be low until 'Trig' is received with EdgeType set to 1b. This means that the Retimer_Force_PWR_Event_Port1 and Retimer_Force_PWR_Event_Port2 GPIO's are left in the state specified for them in GPIO_CONFIG.GPIOData until 'Trig' is received. INTEL_VID_STATUS.RetimerForcePower reflects the state of this virtual GPIO.
			34d (22h)	Fault_Input_Event_Port2: This virtual GPIO is assumed to be high until 'Trig' is received with EdgeType set to 0b.
			33d (21h)	Fault_Input_Event_Port1: This virtual GPIO is assumed to be high until 'Trig' is received with EdgeType set to 0b.
			else	The task will be rejected.
	7:1	Reserved		
	0	EdgeType	0b	Falling edge
			1b	Rising edge
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.			
Task Completion	This task completes successfully once the applicable actions for the GPIO event are executed. The task is rejected if the GPIO event is not a valid input GPIO event. This task should not be issued for a GPIO event that is assigned to a physical GPIO in the GPIO_CONFIG register.			
Side Effects	Depending upon the selected GPIO event various actions may be taken by the PD controller. Refer to the description of each GPIO event in Table 3-94.			
Additional Information	The PD controller executes the event handler for the rising or falling edge as given in the input of this command. Therefore, it is possible to execute consecutive rising edges or consecutive falling edges. Although, in most cases subsequent rising edges will not have any action, it is not recommended that the host issue consecutive 'Trig' tasks with the same EdgeType and GPIO Event #.			

4.8.6 'I2Cr' - I2C read transaction

Table 4-37. 'I2Cr' - Executes I2C read transaction on I2C3m.

Description	**The 'I2Cr' task may be used to cause the PD controller to read from a specified slave address and register offset using a I^{2}C read transaction via the I2C3m_SDA and I2C3m_SCL pins.**		
INPUT DATAX	**Bit**	**Name**	**Description**
	Byte 3: Number of bytes to read from the slave.		
	7:0	NumBytes	
	Byte 2: Register offset to use in the I2C read transaction.		
	7:0	RegisterOffset	
	Byte 1: Slave Address		
	7	Reserved	
	6:0	Slave to use for the transaction.	
OUTPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 2-65: Data Bytes read from the slave (in order received)		
	511:0	Data	
	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	The PD controller completes once it has succesfully read the specified number of bytes, or the I^{2}C transaction terminated for some other reason.		
Side Effects	This task will cause the PD controller to issue a command on the I2C3m port. It could result in INT_EVENTx.I2CMasterNACKed being asserted.		
Additional Information	None		

4.8.7 'I2Cw' - I2C write transaction

Table 4-38. 'I2Cw' - Executes I2C write transaction on I2C3m.

Description	**The 'I2Cw' task may be used to cause the PD controller to write a particular I^{2}C transaction using I2C3m_SDA and I2C3m_SCL.**		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 5-14: Payload for the I2C transaction		
	Byte 4: Register Offset for the I2C transaction		
	7:0	Register offset	
	Bytes 2-3: Length		
	15:8	Reserved	
	7:0	Number of bytes in the transaction payload.	
	Byte 1: Slave Address		
	7	Reserved	
	6:0	Slave to use for the transaction.	
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	The PD controller maintains a queue of transactions to send on the I2C3m port. If the PD controller has been configured to send transactions upon certain events, it is possible there is a transaction in the queue when the 'I2Cw' task is received. In that case the task will complete successfully after the transaction is inserted into the queue. If the PD controller fails to insert the task into the queue for any reason, the task is rejected. Therefore, when this task is completed succesfully it does not guarantee that the I2C transaction is complete. If possible, the host should use the 'I2Cr' 4CC task to confirm the write was successful.		
Side Effects	When successful, this task will cause the PD controller to issue a command on the I2C3m port. This could result in INT_EVENTx.I2CMasterNACKed being asserted.		
Additional Information	If the DATAX register is written with more than 14 bytes, all bytes beyond byte 14 are ignored. The PD controller has a limit on the maximum length of the I^{2}C write transaction.		

4.9 UCSI 4CC Tasks

The 'UCSI' Task is used to pass a UCSI Task from the PPM to the LPM (terms defined in the UCSI specification). The input and output data are dependent on the Task passed. The supported Tasks and their associated data structures are outlined in the sub-sections below. The PD controller implements UCSI version 1.1.

The host should consider the Task Return code of the 4CC Task 'UCSI' to be binary. If the return code is 0h, then it was successful. Otherwise it failed, and the OPM or host can use the 'UCSI' Task with Command=13h (GET_ERROR_STATUS) to obtain detailed reasons for the failure.

Note that when a 'UCSI' Task requires sending a USB PD message while the PD_STATUS.PresentPDRole=0, the PD controller must wait until the source yields the bus before it can transmit the message. Therefore, a minimum response time cannot be guaranteed. The PD controller will wait forever until the source sets Rp = SinkTxOK. In some scenarios, the host may need to issue the 'ABRT' Task if the delay becomes too long.

4.9.1 'UCSI' - PPM_RESET

Table 4-39. 'PPM_RESET' - This Task is used to reset the Platform Policy Manager

Description	PPM_RESET - This Task is used to reset the Platform Policy Manager		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 3-7: Additional Input Data		
	47:7	Reserved	Reserved
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.
	Byte 2: Data Length per UCSI spec (write only as 0x00)		
	Byte 1: Command Code as defined by UCSI: 0x01 (PPM_RESET)		
	7:0	Command:0x01	
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	This Task will always return a successful task return code.		
Side Effects	None		
Additional Information	None		

4.9.2 'UCSI' - CONNECTOR_RESET

Table 4-40. 'CONNECTOR_RESET' - This Task is used to reset the Type-C Port

Description	**CONNECTOR_RESET - This Task is used to reset the Type-C Port**		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 3-8: Additional Input Data		
	47:7	Reserved	Reserved
	6:0	Connector Number	This field indicates the connector to be reset. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.
	Byte 2: Data Length per UCSI spec (write only as 0x00)		
	Byte 1: Command Code as defined by UCSI: 0x03 (CONNECTOR_RESET)		
	7:0	Command:0x03	
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	This Task is only successful if the device is not in Dead battery state or the port is not the Dead Battery Power Provider.		
Side Effects	None		
Additional Information	None		

4.9.3 'UCSI' - GET_CAPABILITY

The PD controller responds to this Task based on the port associated with the slave address used.

Table 4-41. 'GET_CAPABILITY' - This Task is used to get the capabilities of the device

Description	**GET_CAPABILITY - This Task is used to get the capabilities of the device**		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 3-8: Additional Input Data		
	47:7	Reserved	Reserved
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.
	Byte 2: Data Length per UCSI spec (write only as 0x00)		
	Byte 1: Command Code as defined by UCSI: 0x06 (GET_CAPABILITY)		
	7:0	Command:0x06	
OUTPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 11-17: Spec Versions		
	55:40	bcdTypeCVersion	Type-C Version Supported (0x130)
	39:24	bcPDVersion	PD Version Supported (0x030)
	23:8	bcdBCVersion	BC version supported (0x120)
	7:0	Reserved	
	Byte 10: Number of Alternate Modes		
	7:0	bNumAltModes	Not Populated by PD controller
	Bytes 7-9: bmOptionalFeatures		
	23:8	Reserved	
	7	PD Reset notification supported	
	6	External supply notification supported	
	5	Cable details supported	
	4	PDO Details supported	
	3	Alternate mode override supported	
	2	Alternate mode details supported	
	1	SET_POWER_LEVEL supported	
	0	SET_CCOM Supported	
	Byte 6: Number of Connectors		
	7	Reserved	
	6:0	bNumConnectors	This is set to the number of ports supported by the PD controller. However, it not recommended that the host rely on this value since there could be multiple PD controllers in the system.
	Bytes 2-5: bmAttributes		
	31:16	Reserved	
	15:8	bmPowerSource	
	7	Reserved	
	6	USB Type-C Current	
	5:3	Reserved	
	2	USB Power Delivery	
	1	Battery Charging	
	0	Disabled State Support	
	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	This Task will always return a successful task return code.		

Table 4-41. 'GET_CAPABILITY' - This Task is used to get the capabilities of the device (continued)

Description	GET_CAPABILITY - This Task is used to get the capabilities of the device
Side Effects	None
Additional Information	None

4.9.4 'UCSI' - GET_CONNECTOR_CAPABILITY

Table 4-42. 'GET_CONNECTOR_CAPABILITY' - This Task is used to get the capabilities of the port

Description	**GET_CONNECTOR_CAPABILITY - This Task is used to get the capabilities of the port**			
INPUT DATAX	**Bit**	**Name**	**Description**	
	Bytes 3-8: Additional Input Data			
	47:7	Reserved		
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.	
	Byte 2: Data Length per UCSI spec (write only as 0x00)			
	Byte 1: Command Code as defined by UCSI: 0x07 (GET_CONNECTOR_CAPABILITY)			
	7:0	Command:0x07		
OUTPUT DATAX	**Bit**	**Name**	**Description**	
	Bytes 2-3: Connector Capabilities			
	15:14	Reserved		
	13	Swap to SNK	Asserted if swap to sink supported. Only valid if DRP mode is supported.	
	12	Swap to SRC	Asserted if swap to source supported. Only valid if DRP mode is supported.	
	11	Swap to UFP	Asserted if swap to UFP supported. Only valid if DRP or Rp-only is supported.	
	10	Swap to DFP	Asserted if swap to DFP supported. Only valid if DRP or Rd-only is supported.	
	9	Consumer	This bit is set when the system is capable of consuming power from the USB connector. Only valid if DRP mode or Rd-only mode is supported.	
	8	Provider	This bit is set when the system is capable of providing power to the USB connector. Only valid if DRP mode or Rp-only mode is supported.	
	7:0	Operation Mode	Indicate the modes of operation supported.	
			1xxxxxxxb	Alternate Mode
			x1xxxxxxb	USB3
			xx1xxxxxb	USB2
			xxx1xxxxb	Debug Accessory
			xxxx1xxxb	Analog Audio Accessory
			xxxxx1xxb	DRP (Rp/Rd)
			xxxxxx1xb	Rd Only
			xxxxxxx1b	Rp only
	Byte 1: Standard Task Return Code. See also Table 4-1.			
Task Completion	This Task will always return a successful task return code.			
Side Effects	None			
Additional Information	None			

4.9.5 'UCSI' - SET_CCOM

Table 4-43. 'SET_CCOM' - This Task is used to set the CC operation mode that the OPM wants the connector to operate at for the current connection.

Description	**SET_CCOM - This Task is used to set the CC operation mode that the OPM wants the connector to operate at for the current connection.**			
INPUT DATAX	**Bit**	**Name**	**Description**	
	Bytes 3-8: Additional Input Data			
	47:10	Reserved		
	9:7	CC Operation Role	This field must be a non-zero value.	
			xx1b	If this bit is set the connector shall operate only in Rp mode (Source only).
			x1xb	If this bit is set the connector shall operate only in Rd mode (Sink only).
			1xxb	If this bit is set then the connector shall operate as a dual-role port (DRP).
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.	
	Byte 2: Data Length per UCSI spec (write only as 0x00)			
	Byte 1: Command Code as defined by UCSI: 0x08 (SET_CCOM)			
	7:0	Command:0x08		
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.			
Task Completion	This Task is successful if port is already in the role it is requested for or after changing to the requested CC Operation Role.			
Side Effects	None			
Additional Information	None			

4.9.6 'UCSI' - SET_UOR

Table 4-44. 'SET_UOR' - This Task is used to set the USB operation role that the OPM wants the connector to operate at for the current connection

Description	**SET_UOR - This Task is used to set the USB operation role that the OPM wants the connector to operate at for the current connection**			
INPUT DATAX	**Bit**	**Name**	**Description**	
	Bytes 3-8: Additional Input Data			
	47:10	Reserved		
	9:7	USB Operation Role	xx1b	If this bit is set the connector shall initiate swap to DFP if not already in the DFP role
			x1xb	If this bit is set the connector shall initiate swap to UFP if not already in the UFP role
			1xxb	If this bit is set then the connector shall accept role swap change requests from the port partner. If this bit is cleared then the connector shall reject role swap change requests from the port partner
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.	
	Byte 2: Data Length per UCSI spec (write only as 0x00)			
	Byte 1: Command Code as defined by UCSI: 0x09 (SET_UOR)			
	7:0	Command:0x09		
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.			
Task Completion	This Task is successful if port is already in the role it is requested for or after successful completion of Data Role Swap Message if not in role requested for.			
Side Effects	This Task may result in the PD controller sending a DR_Swap USB PD message.			
Additional Information	None			

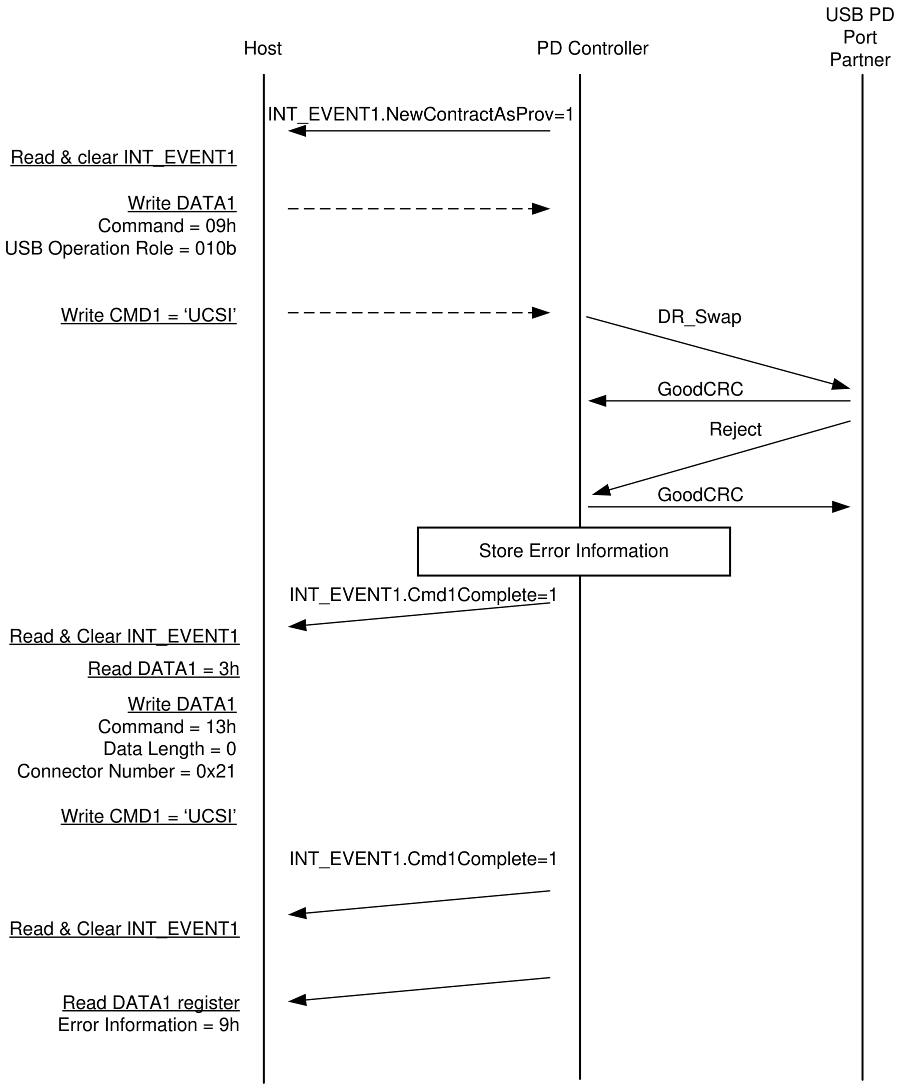


Figure 4-6. Example of an unsuccessful SET_UoR command.

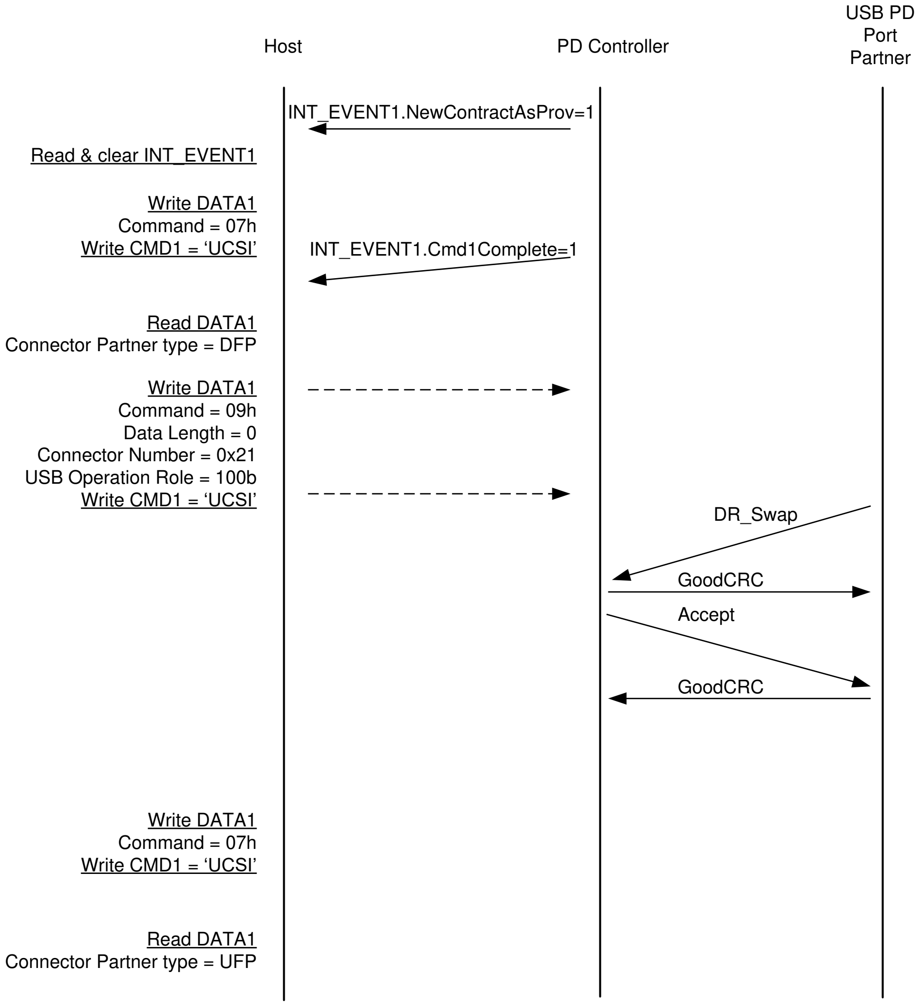


Figure 4-7. Example of successful SET_UoR command.

4.9.7 'UCSI' - SET_PDR

Table 4-45. 'SET_PDR' - This Task is used to set the power direction that the OPM wants the connector to operate at for the current connection

Description	**SET_PDR - This Task is used to set the power direction that the OPM wants the connector to operate at for the current connection**			
INPUT DATAX	**Bit**	**Name**	**Description**	
	Bytes 3-8: Additional Input Data			
	47:10	Reserved		
	9:7	Power Direction Role	xx1b	If this bit is set the connector shall initiate swap to Source if not already in the Source role
			x1xb	If this bit is set the connector shall initiate swap to Sink if not already in the Sink role
			1xxb	If this bit is set then the connector shall accept power swap change requests from the port partner. If this bit is cleared then the connector shall reject power swap change requests from the port partner
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.	
	Byte 2: Data Length per UCSI spec (write only as 0x00)			
	Byte 1: Command Code as defined by UCSI: 0x0B (SET_PDR)			
	7:0	Command:0x0B		
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.			
Task Completion	This Task is successful if port is already in the role it is requested for or after successful completion of Power Role Swap Message if not in role requested for.			
Side Effects	This Task may result in the PD controller sending a PR_Swap USB PD message.			
Additional Information	None			

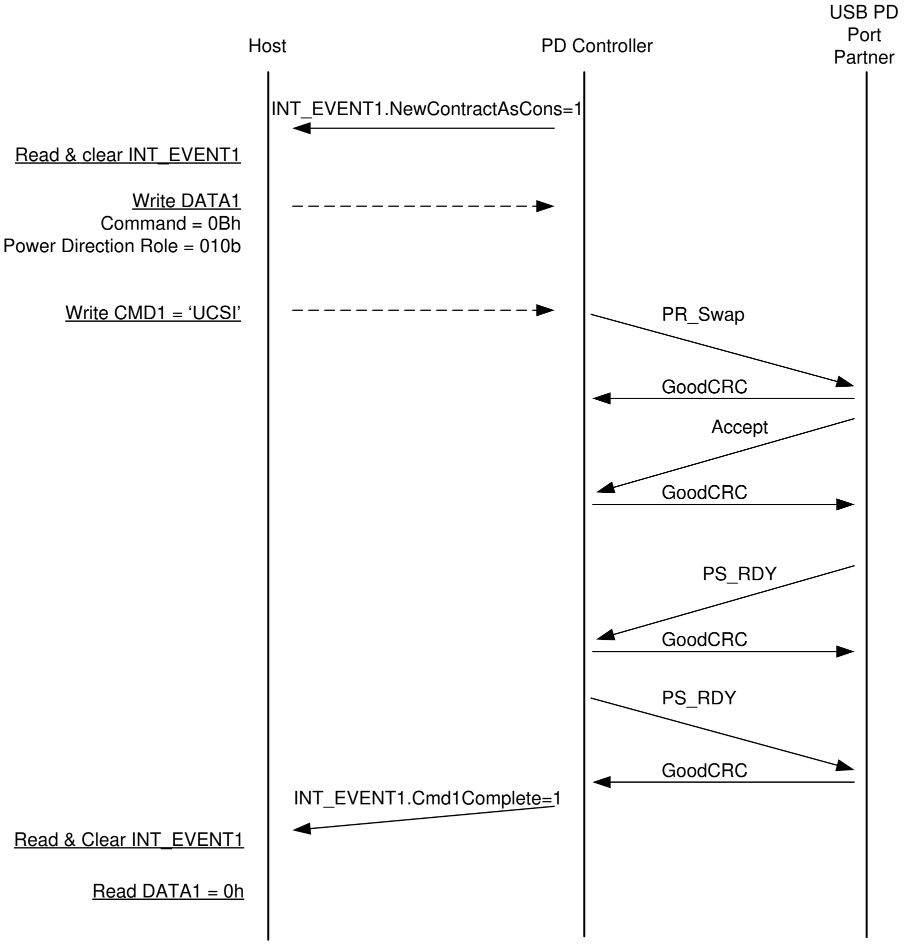


Figure 4-8. Example of successful SET_PDR command.

4.9.8 'UCSI' - GET_ALTERNATE_MODES

Table 4-46. 'GET_ALTERNATE_MODES' - This Task is used to get the Alternate Modes that the Connector/Cable/Attached Device is capable of supporting

Description	**GET_ALTERNATE_MODES - This Task is used to get the Alternate Modes that the Connector/Cable/Attached Device is capable of supporting**		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 3-8: Additional Input Data		
	47:26	Reserved	
	25:24	Number of Alternate Modes	Number of Alternate Modes to return starting from the Alternate Mode Offset. The number of Alternate Modes to return is the value in this field plus 1. The maximum value of this field is 1.
	23:16	Alternate Mode Offset	Starting offset of the first alternate mode to be returned
	15	Reserved	
	14:8	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.
	7:3	Reserved (Write 0)	
	2:0	Recipient	00b Port (Connector) 01b SOP 10b SOP' 11b SOP"
	Byte 2: Data Length per UCSI spec (write only as 0x00)		
	Byte 1: Command Code as defined by UCSI: 0x0C (GET_ALTERNATE_MODES)		
OUTPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 11-14: Second Mode VDO		
	31:0	MID[1]	Mode ID associated with the below SVID
	Bytes 9-10: Second SVID		
	15:0	SVID[1]	Standard or Vendor ID
	Bytes 5-8: First Mode VDO		
	31:0	MID[0]	Mode ID associated with the below SVID
	Bytes 3-4: First SVID		
	15:0	SVID[0]	Standard or Vendor ID
	Byte 2: Number of Data Bytes returned		
	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	If a value greater than 11b is written to Recipient in Input DATAX, if Number of Alternate Modes is greater than 1 in Input DATAX, or the Alternate Mode Offset is greater than the maximum allowed number of total modes allowed by the USB-PD spec in Input DATAX, then this task will be rejected. Otherwise, this Task will complete successfully.		
Side Effects	None		
Additional Information	None		

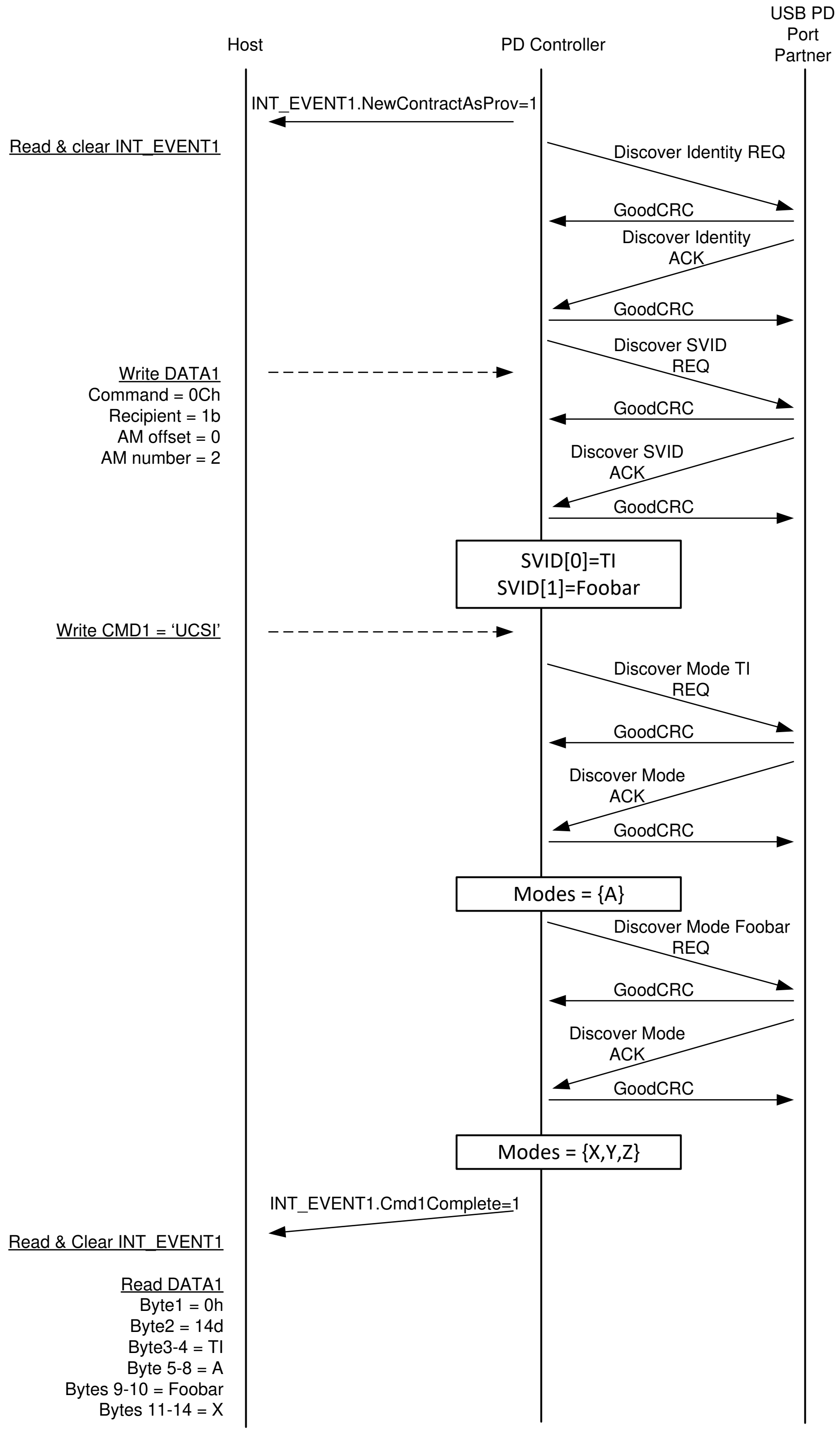


Figure 4-9. Example of successful GET_ALTERNATE_MODES command.

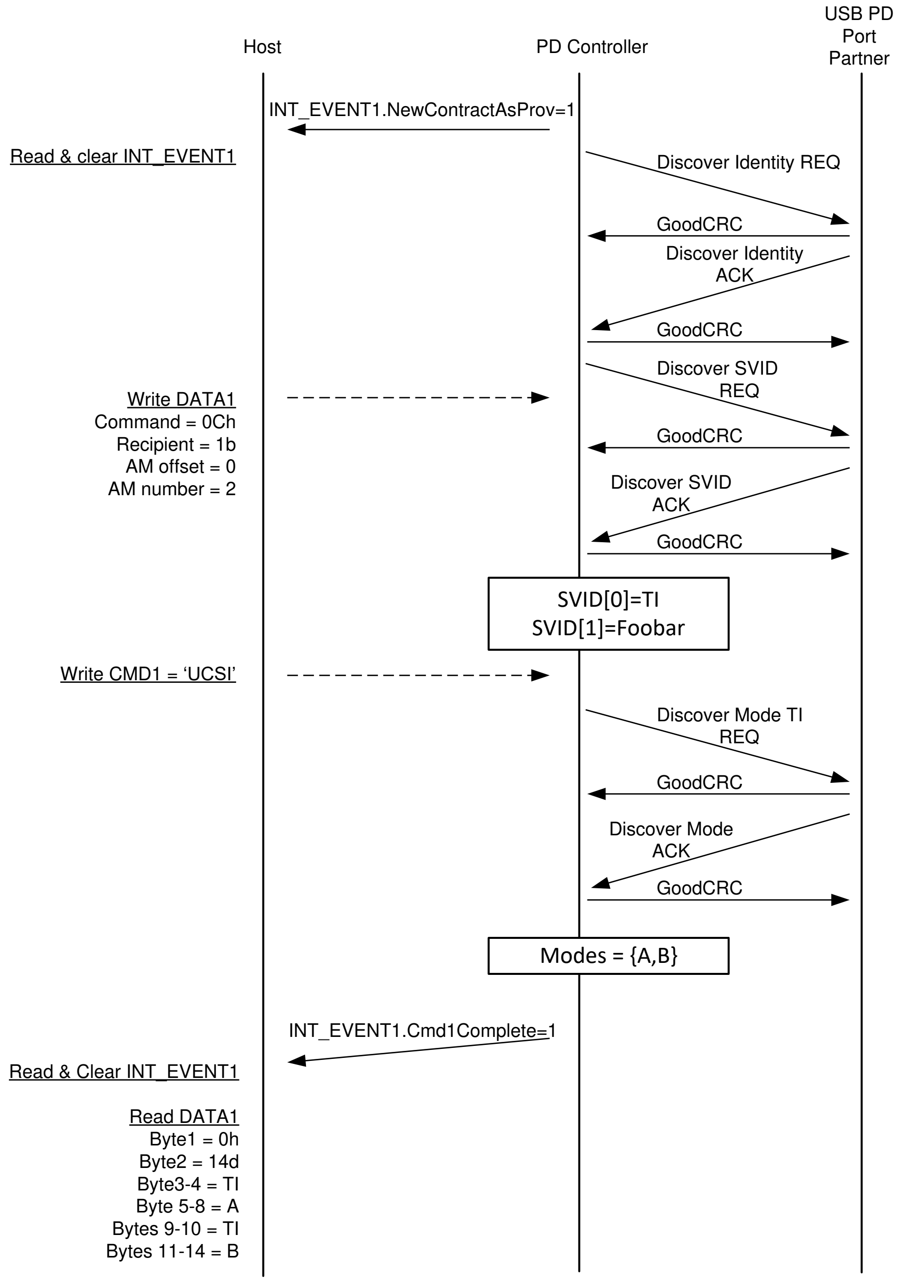


Figure 4-10. Example of a successful GET_ALTERNATE_MODES command.

4.9.9 'UCSI' - GET_CAM_SUPPORTED

Table 4-47. 'GET_CAM_SUPPORTED' - This Task is used to get the list of Alternate Modes that are currently supported on the connector. These are the alternate modes supported by the PD controller and the connected cable or Port Partner.

Description	**GET_CAM_SUPPORTED - This Task is used to get the list of Alternate Modes that are currently supported on the connector. These are the alternate modes supported by the PD controller and the connected cable or Port Partner.**		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 3-8: Additional Input Data		
	47:7	Reserved	Reserved
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.
	Byte 2: Data Length per UCSI spec (write only as 0x00)		
	Byte 1: Command Code as defined by UCSI: 0x0D (GET_CAM_SUPPORTED)		
	7:0	Command:0x0D	
OUTPUT DATAX	**Bit**	**Name**	**Description**
	Byte 2: Supported Alternate Modes		
	7:0	bmAltModeSupported	A value of 1 in the bit position corresponding to an Alt. Mode indicates that it is supported. The list of Alt. Modes is available from GET_ALTERNATE_MODES UCSI command.
	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	This Task will always return a successful task return code.		
Side Effects	None		
Additional Information	None		

4.9.10 'UCSI' - GET_CURRENT_CAM

Table 4-48. 'GET_CURRENT_CAM' - This Task is used to get the current Alternate Modes that the connector is currently operating in.

Description	GET_CURRENT_CAM - This Task is used to get the current Alternate Modes that the connector is currently operating in.		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 3-8: Additional Input Data		
	47:7	Reserved	Reserved
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.
	Byte 2: Data Length per UCSI spec (write only as 0x00)		
	Byte 1: Command Code as defined by UCSI: 0x0E (GET_CURRENT_CAM)		
	7:0	Command:0x0E	
OUTPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 2-64: Supported Alternate Modes		
	503:16	CurrentAlternateModeN	For each Alternate Mode that has been entered and is presently active, there is a byte CurrentAlternateModeN.
	15:8	CurrentAlternateMode2	Second offset into the list of Alternate Modes supported that the connector is currently operating in.
	7:0	CurrentAlternateMode1	First offset into the list of Alternate Modes supported that the connector is currently operating in.
	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	This Task will always return a successful task return code.		
Side Effects	None		
Additional Information	None		

4.9.11 'UCSI' - SET_NEW_CAM

Table 4-49. 'SET_NEW_CAM' - This Task is used to set the Alternate Mode that the OPM wants the connector to operate in for this present connection.

Description	**SET_NEW_CAM - This Task is used to set the Alternate Mode that the OPM wants the connector to operate in for this present connection.**		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 3-8: Additional Input Data		
	47:16	Reserved	Reserved
	15:8	New CAM	Offset into the list of supported alternate modes for the one the OPM wants the connector to operate under.
	7	EnterOrExit	Set this bit to 1 to Enter the specified alternate mode (New CAM), and set it to 0 to Exit the specified alternate mode.
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.
	Byte 2: Data Length per UCSI spec (write only as 0x00)		
	Byte 1: Command Code as defined by UCSI: 0x0F (SET_NEW_CAM)		
	7:0	Command:0x0F	
OUTPUT DATAX	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	This Task will only return a successful task code after it has entered or exited the requested mode.		
Side Effects	None		
Additional Information	None		

4.9.12 'UCSI' - GET_PDOS

Table 4-50. 'GET_PDOS' - This Task is used to get the Sink or Source PDOs associated with the identified connector

Description	GET_PDOS - This Task is used to get the Sink or Source PDOs associated with the identified connector			
INPUT DATAX	**Bit**	**Name**	**Description**	
	Bytes 3-8: Additional Input Data			
	47:21	Reserved	Reserved	
	20:19	Source Capabilities Type	Type of Source Capabilities requested.	
			00b	Current Supported Source Capabilites
			01b	Advertised Capabilities
			10b	Maximum Supported Source Capabilities
			11b	Not Used
	18	Source or Sink PDOs	This field shall be set to one if the OPM wants to retrieve the Source PDOs otherwise it wants to retrieve the Sink PDOs.	
	17:16	Number of PDOs	Number of PDOs to return starting from the PDO Offset. The number of PDOs to return is the value in this field plus 1.	
	15:8	PDO Offset	Starting offset of the first PDO to be returned	
	7	Partner PDO	This field shall be set to one if the OPM wants to retrieve the PDOS of the device attached to the connector	
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.	
	Byte 2: Data Length per UCSI spec (write only as 0x00)			
	Byte 1: Command Code as defined by UCSI: 0x10 (GET_PDOS)			
	7:0	Command:0x10		
OUTPUT DATAX	**Bit**	**Name**	**Description**	
	Bytes 3-18: PDO's at the given offset			
	127:96	PDO4	Fourth PDO at offset.	
	95:64	PDO3	Third PDO at offset.	
	63:32	PDO2	Second PDO at offset.	
	31:0	PDO1	First PDO at offset.	
	Byte 2: Number of bytes reported. This is the number of PDO's to be reported times 4.			
	Byte 1: Standard Task Return Code. See also Table 4-1.			
Task Completion	If Partner PDO is set to 1, then this task will successful once the requested capabilities information is obtained from the Port Partner or if that information had already been obtained. If Partner PDO is set to 0 and Source Capabilities Type is set to 11b, then this task will be rejected.			
Side Effects	This could result in the PD controller sending a Get_Source_Cap USB PD message and re-negotiating the PD contract. It could also result in the PD controller sending a Get_Sink_Cap USB PD message.			
Additional Information	None			

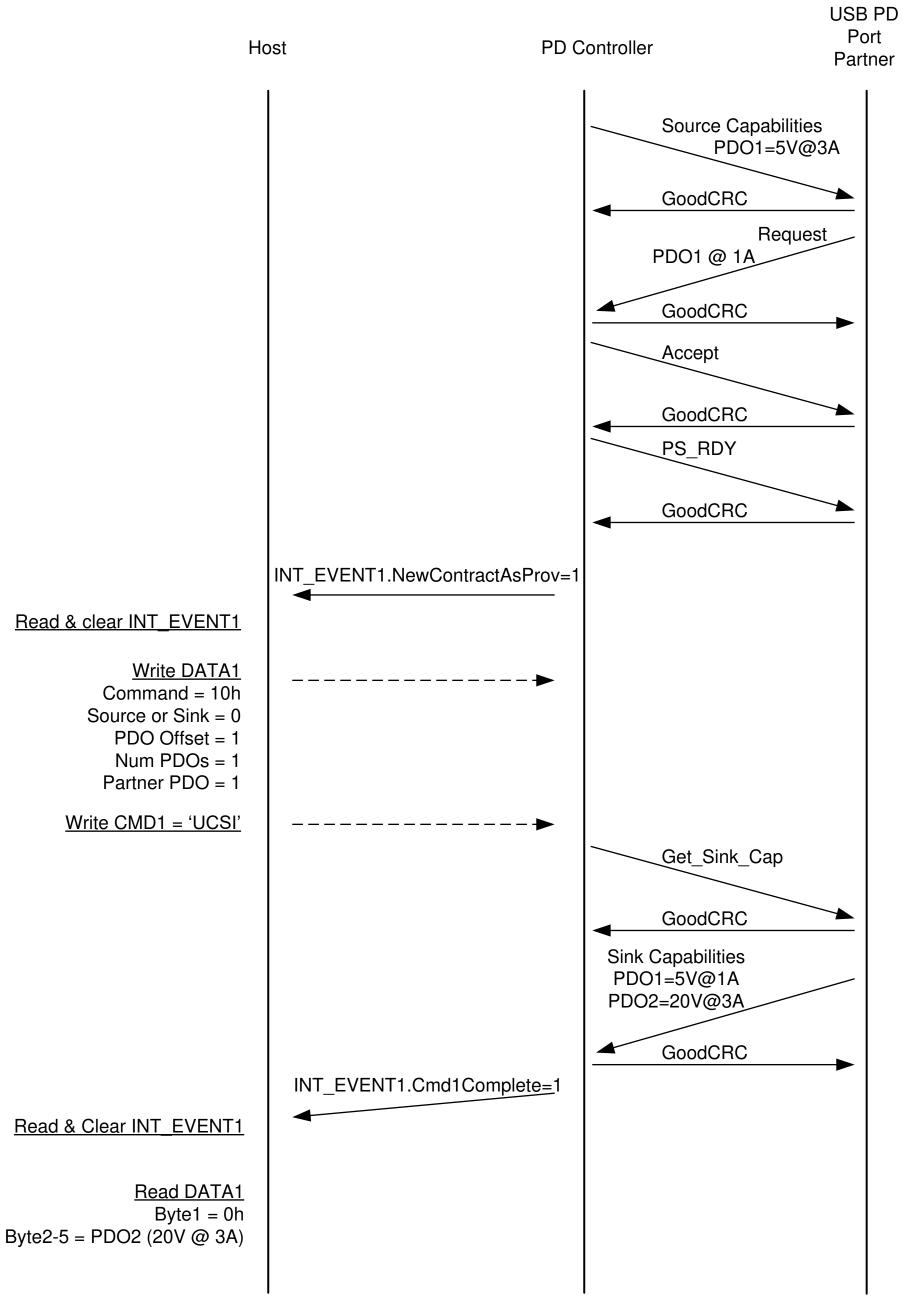


Figure 4-11. Example of a successful GET_PDOS command.

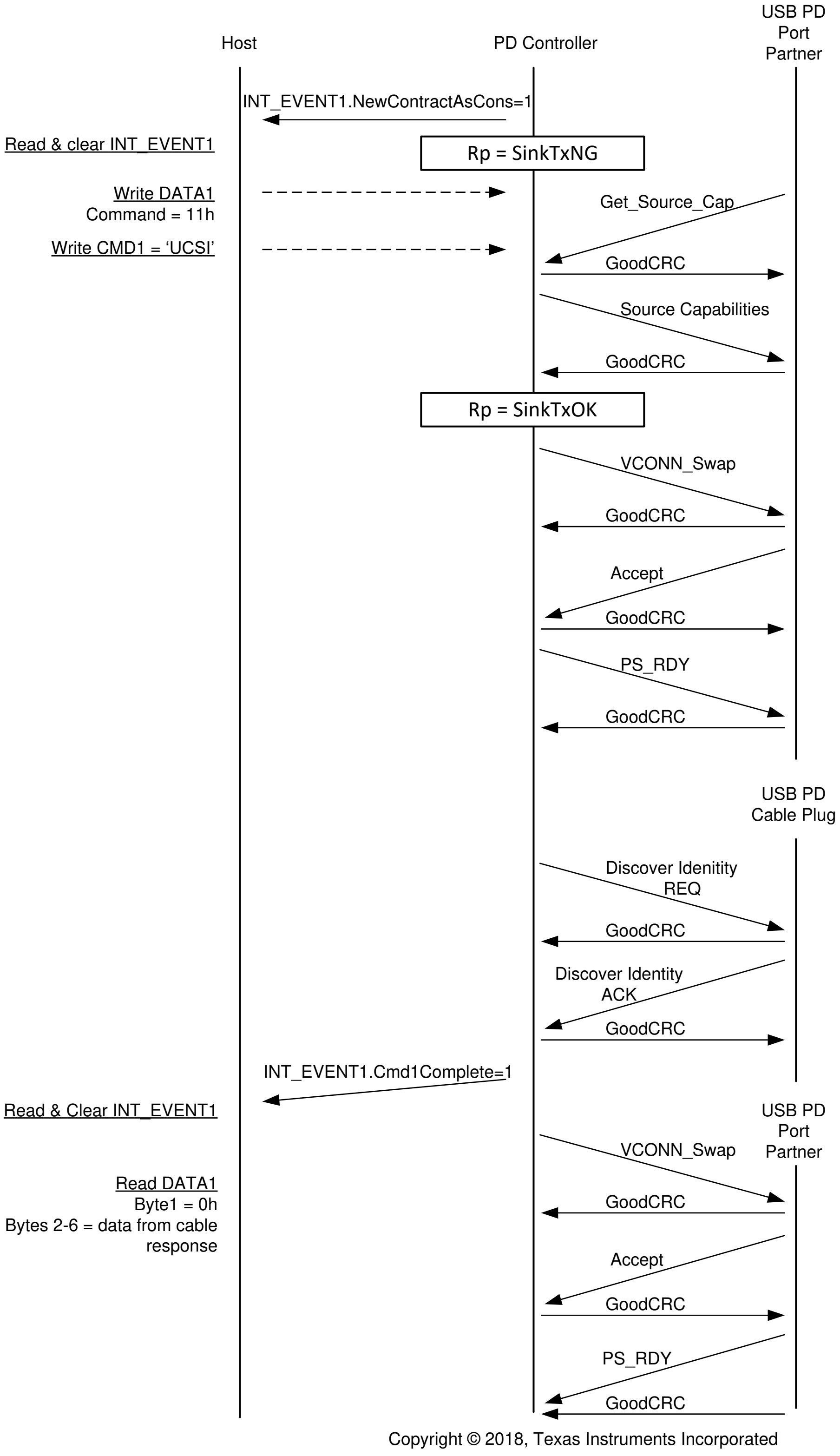
4.9.13 'UCSI' - GET_CABLE_PROPERTY

Table 4-51. 'GET_CABLE_PROPERTY' - This Task is used to get the properties of the cable attached to the identified connector.

Description	**GET_CABLE_PROPERTY - This Task is used to get the properties of the cable attached to the identified connector.**		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 3-8: Additional Input Data		
	47:8	Reserved	Reserved
	7	Reserved	
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.
	Byte 2: Data Length per UCSI spec (write only as 0x00)		
	Byte 1: Command Code as defined by UCSI: 0x11 (GET_CABLE_PROPERTY)		
	7:0	Command:0x11	
OUTPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 4-6: Other Parameters		
	23:20	Reserved	
	19:16	Latency	This is a parameter directly from the USB PD specification.
	15:14	Reserved	
	13	ModalSupport	This bit will only be supported if this is an active cable that supports at least one alternate mode. The OPM may use the GET_ALTERNATE_MODE command to query which alternate modes the cable supports.
	12:11	PlugEndType	Type of plug on the cable connected. 00b USB Type-A 01b USB Type-B 10b USB Type-C 11b Other (non USB)
	10	Directionality	This bit is only asserted if the cable lane directionality is configurable.
	9	CableType	If the cable is active this bit will be 1, if it is passive this bit will be 0.
	8	VbusInCable	If the cable has a VBUS wire from end-to-end this bit will be asserted.
	7:0	CurrentCapability	The amount of current the cable can carry in 50 mA steps.
	Bytes 2-3: SpeedParameters		
	15:2	SpeedMantissa	This field defines the mantissa to be applied to the Speed Exponent to calculate the speed.
	1:0	SpeedExponent	This field defines the base 10 exponent times 3, that is applied to the speed mantissa when calculating bit rate that the cable supports. 00b per second. 01b Kb/s 10b Mb/s 11b Gb/s
	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	This task will be successful if the PD controller is able to send a query to the cable and it responds in a timely manner.		
Side Effects	This could cause the PD controller to send a VCONN_Swap to become the VCONN Provider. It may also result in the PD controller sending a Discover Identity Request message to the cable plug.		

Table 4-51. 'GET_CABLE_PROPERTY' - This Task is used to get the properties of the cable attached to the identified connector. (continued)

Description	GET_CABLE_PROPERTY - This Task is used to get the properties of the cable attached to the identified connector.
Additional Information	None



Copyright © 2018, Texas Instruments Incorporated

Figure 4-12. Example of successful GET_CABLE_PROPERTY command.

4.9.14 'UCSI' - GET_CONNECTOR_STATUS

When this 4CC task is executed by the host, and the PD controller has placed the output data in the DATAx register, the PD controller clears the internal connector status. So if the host executes this 4CC task a second time it will read all zeros, unless a new event has occured since the last read.

Table 4-52. 'GET_CONNECTOR_STATUS' - This Task is used to get the capabilities of the device

Description	**GET_CONNECTOR_STATUS - This Task is used to get the capabilities of the device**		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 3-8: Additional Input Data		
	47:7	Reserved	Reserved
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.
	Byte 2: Data Length per UCSI spec (write only as 0x00)		
	Byte 1: Command Code as defined by UCSI: 0x12 (GET_CONNECTOR_STATUS)		
	7:0	Command:0x12	

Table 4-52. 'GET_CONNECTOR_STATUS' - This Task is used to get the capabilities of the device (continued)

Description	GET_CONNECTOR_STATUS - This Task is used to get the capabilities of the device			
OUTPUT DATAX	Bit	Name	Description	
	Bytes 4-10: Additional Connector Status Information			
	55:52	Reserved		
	51:50	Provider Capabilities Limited Reason	00b	Power Budget Lowered
			01b	Reaching Power Budget Limit
			10b	Reserved
			11b	Reserved
	49:48	Battery Charging Capability Status	00b	No Charging
			01b	Nominal Charging Rate
			10b	Slow Charging Rate
			11b	Very Slow Charging Rate
	47:16	Request Data Object (RDO)	This field is only valid when the Connect Status field is set to one and the Power Operation Mode field is set to PD	
	15:13	Connector Partner Type	000b	Reserved
			001b	DFP Attached
			010b	UFP Attached
			011b	Powered cable/No UFP Attached
			100b	Powered cable/ UFP Attached
			101b	Debug Accessory Attached
			110b	Audio Adapter Accessory attached
			111b	Reserved
	12:5	Connector Partner Flags	xxxxxxx1b	USB Mode
			xxxxxx1xb	Alternate Mode
			else	Reserved
	4	Power Direction	0b	Connector is operating as a SINK
			1b	Connector is operating as a SOURCE
	3	Connect Status	Set to 1 when a device is connected	
	2:0	Power Operation Mode	000b	Reserved
			001b	USB Default Operation
			010b	Battery Charging
			011b	Power Delivery
			100b	USB Type-C Current (1.5A)
			101b	USB Type-C Current (3.0A)
			110b	Reserved
			111b	Reserved

Table 4-52. 'GET_CONNECTOR_STATUS' - This Task is used to get the capabilities of the device (continued)

Description	GET_CONNECTOR_STATUS - This Task is used to get the capabilities of the device		
OUTPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 2-3: Connector Status Change		
	15	Error	
	14	Connect Change	
	13	Reserved	
	12	Power Direction Change	
	11	Connector Partner Change	
	10	Reserved	
	9	Battery Charging Status Change	
	8	Supported CAM Change	
	7	PD Reset Complete	
	6	Negotiated Power Level Change	
	5	Supported Provider Capabilities Change	
	4	Reserved	
	3	Reserved	
	2	Power Operation Mode Change	
	1	External Supply Change	
	0	Reserved	
	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	This Task will always return a successful task return code.		
Side Effects	None		
Additional Information	None		

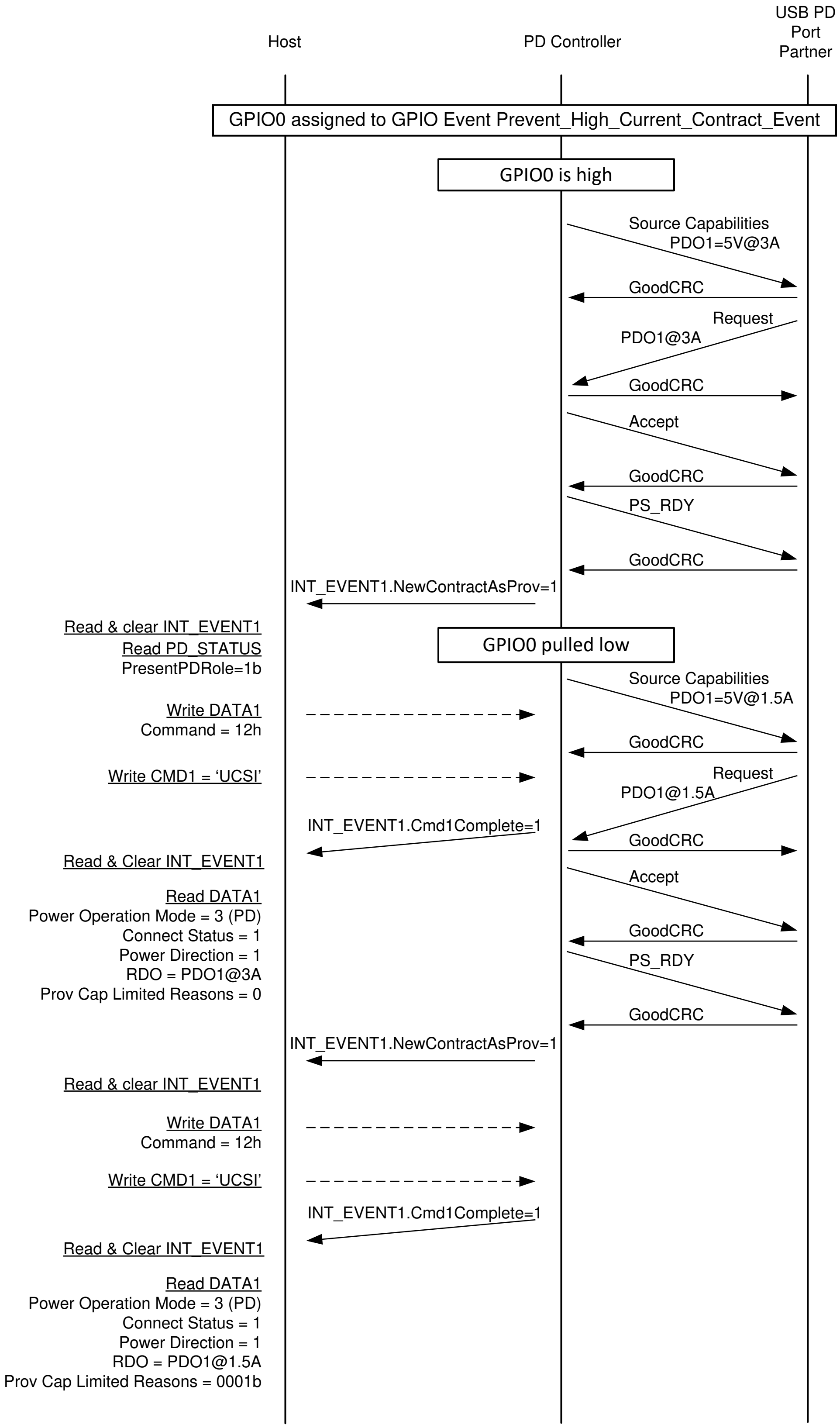


Figure 4-13. Example of successful GET_CONNECTOR_STATUS command.

4.9.15 'UCSI' - GET_ERROR_STATUS

Table 4-53. 'GET_ERROR_STATUS' - This Task is used to get details about an error, if one is reported by the PPM

Description	GET_ERROR_STATUS - This Task is used to get details about an error, if one is reported by the PPM		
INPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 3-8: Additional Input Data		
	47:7	Reserved	Reserved
	6:0	Connector Number	This field indicates the connector whose capabilities are to be retrieved. Write a 0x1 for connector/port 1 and write a 0x2 for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.
	Byte 2: Data Length per UCSI spec (write only as 0x00)		
	Byte 1: Command Code as defined by UCSI: 0x13 (GET_ERROR_STATUS)		
	7:0	Command:0x13	
OUTPUT DATAX	**Bit**	**Name**	**Description**
	Bytes 3-4: Error Information		
	15:13	Reserved	
	12	Swap rejected	
	11	PPM Policy Conflict	
	10	Hard Reset	
	9	Port Partner rejected swap	
	8	Undefined	
	7	overcurrent	
	6	Contract negotiation failure	
	5	Command unsuccessful due to dead battery condition	
	4	CC communication error	
	3	Incompatible Connector Partner	
	2	Invalid command specific parameters	
	1	Non-existant connector	
	0	Unrecognized Command	
	Byte 2: Data Length		
	Byte 1: Standard Task Return Code. See also Table 4-1.		
Task Completion	This Task will always return a successful task return code.		
Side Effects	None		
Additional Information	None		

4.9.16 'UCSI' - SET_POWER_LEVEL

The SET_POWER_LEVEL UCSI command allows the host to set the maximum power that is advertised or consumed. However, there are some usage limitations that the host should avoid.

- The value of GLOBAL_SYSTEM_CONFIG.EnableSPM should not be 1b, while the SET_POWER_LEVEL command is active, that could cause a conflict between the two mechanisms for limiting advertised power.
- The "USB Type-C Current" field in this command has no meaning to the PD controller in the sink mode (when PD_STATUS.PresentPDRole = 0). The PD controller does not control how much power the system is consuming, therefore it cannot take any action.
- The SET_POWER_LEVEL command works by modifying the TX_SINK_CAPS or TX_SOURCE_CAPS register, depending upon the value of the SourceOrSink field. Therefore, the host cannot modify the TX_SINK_CAPS or TX_SOURCE_CAPS while the SET_POWER_LEVEL command is active.
- Prior to activating the SET_POWER_LEVEL command, the host must ensure that AUTO_NEGOTIATE_SINK.AutoComputeSinkMinPower is set to 1b.

The SET_POWER_LEVEL command is activated or de-activated through the "USB PD Max Power" field.

Table 4-54. 'SET_POWER_LEVEL' - This Task is used to set the power level the OPM wants to sink or source for the present connection.

<table>
<tr><th>Description</th><th colspan="4">SET_POWER_LEVEL - This Task is used to set the power level the OPM wants to sink or source for the present connection.</th></tr>
<tr><td rowspan="15"><b>INPUT DATAX</b></td><td><b>Bit</b></td><td><b>Name</b></td><td colspan="2"><b>Description</b></td></tr>
<tr><td colspan="4">Bytes 4-8: Power Configuration</td></tr>
<tr><td>39:10</td><td>Reserved</td><td colspan="2"></td></tr>
<tr><td rowspan="5">9:8</td><td rowspan="5">USBTypeCCurrent</td><td colspan="2">This sets the maximum Type-C current to source or sink. This only has an effect on the PD controller if it sets a lower value than PORT_CONTROL.TypeCCurrent.</td></tr>
<tr><td>00b</td><td>The PD controller uses settings from TX_SINK_CAPS.</td></tr>
<tr><td>01b</td><td>3A</td></tr>
<tr><td>10b</td><td>1.5A</td></tr>
<tr><td>11b</td><td>USB Type-C Defaut</td></tr>
<tr><td>7:0</td><td>USBPDMaxPower</td><td colspan="2">Maximum power in 0.5 W units. If set to 0, the PD controller will determine the max power based on TX_SINK_CAPS and AUTO_NEGOTIATE_SINK registers. The host can de-activate the SET_POWER_LEVEL command by writing this field to 0x00 or 0xFF. If it writes this field to any other value, the SET_POWER_LEVEL command is active; even after a disconnect. The PD controller will only advertise the power in this field if it is lower than the power in TX_SINK_CAPS (if SourceOrSink=0) and TX_SOURCE_CAPS (if SourceOrSink=1).</td></tr>
<tr><td colspan="4">Bytes 2-3: Basic Configuration</td></tr>
<tr><td>15</td><td>SourceOrSink</td><td colspan="2">Set this bit to 1 to configure source power limit, and set it to 0 to configure sink power limit.</td></tr>
<tr><td>14:8</td><td>ConnectorNumber</td><td colspan="2">This field indicates the connector whose capabilities are to be retrieved. Write a 0x1h for connector/port 1 and write a 0x2h for connector/port 2. The connector number needs to correspond with the port I2C address the 'UCSI' 4CC Task is written to. A value of zero in this field is illegal.</td></tr>
<tr><td>7:0</td><td>DataLength</td><td colspan="2">Always write as 0x00</td></tr>
<tr><td colspan="4">Byte 1: Command Code as defined by UCSI: 0x14 (SET_POWER_LEVEL)</td></tr>
<tr><td>7:0</td><td>Command:0x14</td><td colspan="2"></td></tr>
<tr><td><b>OUTPUT DATAX</b></td><td colspan="4">Byte 1: Standard Task Return Code. See also Table 4-1.</td></tr>
<tr><td><b>Task Completion</b></td><td colspan="4">This Task will only return a successful task code after it has successfully changed the power to meet the request.</td></tr>
<tr><td><b>Side Effects</b></td><td colspan="4">None</td></tr>
</table>

Table 4-54. 'SET_POWER_LEVEL' - This Task is used to set the power level the OPM wants to sink or source for the present connection. (continued)

Description	SET_POWER_LEVEL - This Task is used to set the power level the OPM wants to sink or source for the present connection.
Additional Information	None

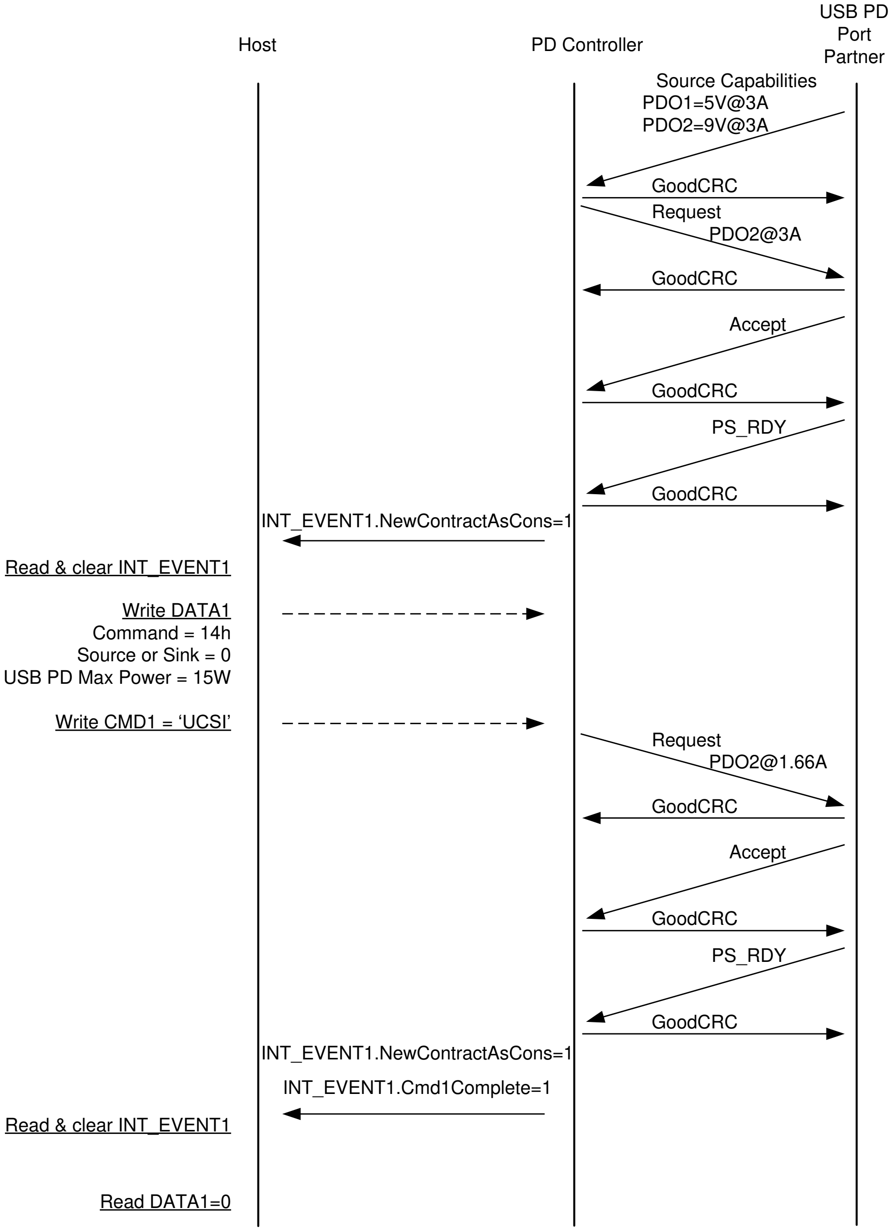


Figure 4-14. Example of successful SET_POWER_LEVEL command as a sink.

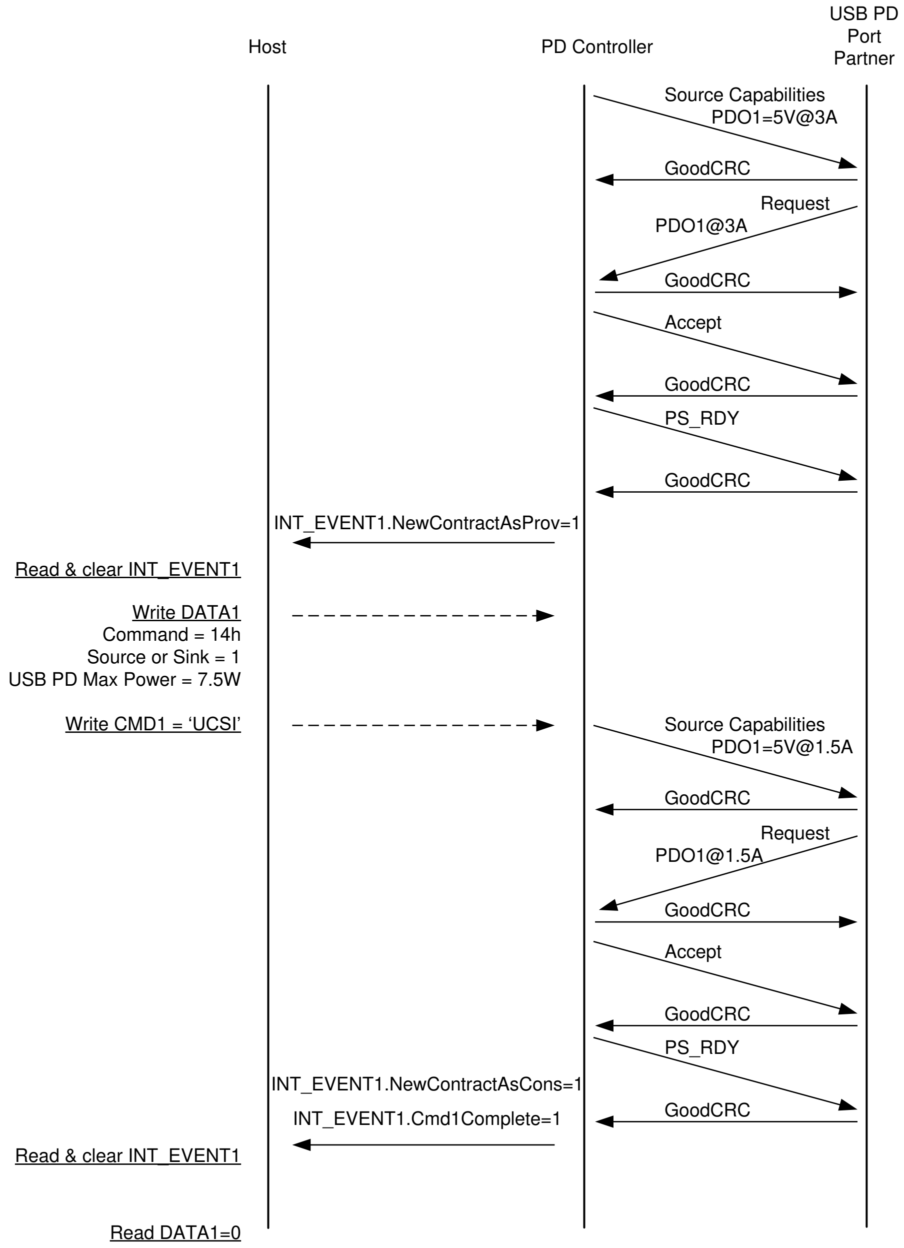


Figure 4-15. Example of a successful SET_POWER_LEVEL command as a source.

Chapter 5
SLVUBT3–May 2020

User Reference

5.1 PD Controller Application Customization

The PD Controller application binary may be pushed over I2C using the I2C_EC port, or the PD controller can read it from an external EEPROM at slave address 0x50 on the I2C3m port. The PD Controller application binary provides a way to customize and initialize the settings of the PD Controller. It allows for any register bit accessible via the Host Interface to be changed *before* the PD Controller application starts normal operation, to configure system-related settings that must be correct before any application decision is made. TI provides a GUI tool to create the PD Controller application binary.

5.2 Loading a Patch Bundle

The patch bundle may contain Application Customization data and a Patch binary that modifies the default application firmware in the PD controller. This section will describe how the host can load the patch bundle. The host uses the I2C_EC bus for all transactions related to loading the patch bundle. As noted in the flow diagram below, the I2C slave address varies depending upon which mode the PD controller is in. The Patch Burst Mode allows the host to push the Patch Bundle to multiple PD controllers simultaneously.

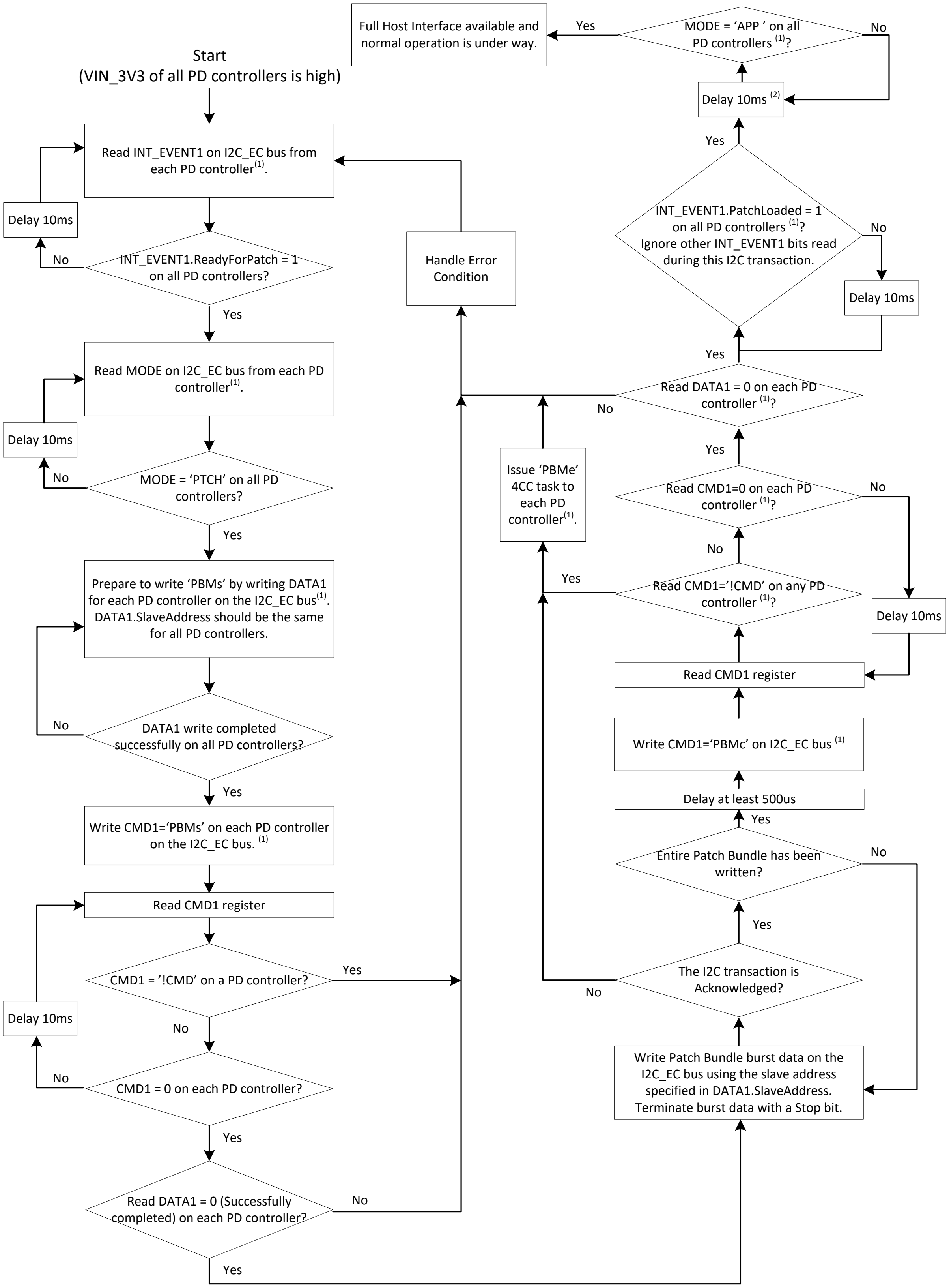
The following flow diagram illustrates the normal successful patch loading process. Other error handling steps may be necessary depending upon the nature of the errors encountered for a particular system. The EC may reset and restart the patch process by issuing a 'PBMe' 4CC Task.

The PD controller can listen for I2C transactions on two unique slave addresses on each I2C port. The following table summarizes the slave addresses in the different modes of operation. GLOBAL_SYSTEM_CONFIG for the I2C2s bus.

Table 5-1. Usage of slave addresses during different modes of operation.

MODE register read-back value	I2C_EC		I2C2s	
	Slave Address #1	Slave Address #2	Slave Address #1	Slave Address #2
'BOOT'	As configured by ADCINx pins for Port A. This is the "Fundamental" I2C slave address.	As configured by ADCINx pins for Port B.	N/A, this port should not be used.	N/A, this port should not be used.
'PTCH'[1]		As specified in the 'PBMs' DATAX input.		
'APP '[2]		As configured by ADCINx pins for Port B.	As specified in Patch Bundle Application Customization for GLOBAL_SYSTEM_CONFIG.Port1I2C2sSlaveAddress.	As specified in Patch Bundle Application Customization for GLOBAL_SYSTEM_CONFIG.Port2I2C2sSlaveAddress.

[1] A successful 'PBMs' Task puts the PD controller into the 'PTCH' mode.
[2] A successful 'PBMc' Task puts the PD controller into the 'APP ' mode.

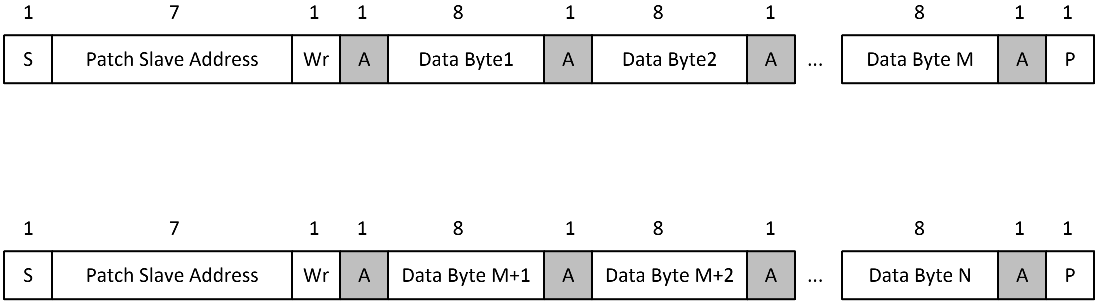

Start
(VIN_3V3 of all PD controllers is high)
Read INT_EVENT1 on I2C_EC bus from each PD controller(1).
Delay 10ms
No
INT_EVENT1.ReadyForPatch = 1 on all PD controllers?
Yes
Read MODE on I2C_EC bus from each PD controller(1).
Delay 10ms
No
MODE = 'PTCH' on all PD controllers?
Yes
Prepare to write 'PBMs' by writing DATA1 for each PD controller on the I2C_EC bus(1). DATA1.SlaveAddress should be the same for all PD controllers.
No
DATA1 write completed successfully on all PD controllers?
Yes
Write CMD1='PBMs' on each PD controller on the I2C_EC bus. (1)
Read CMD1 register
Delay 10ms
CMD1 = '!CMD' on a PD controller?
Yes
No
No
CMD1 = 0 on each PD controller?
Yes
Read DATA1 = 0 (Successfully completed) on each PD controller?
No
Yes
Handle Error Condition
Full Host Interface available and normal operation is under way.
Yes
MODE = 'APP ' on all PD controllers (1)?
No
Delay 10ms (2)
Yes
INT_EVENT1.PatchLoaded = 1 on all PD controllers (1)? Ignore other INT_EVENT1 bits read during this I2C transaction.
No
Delay 10ms
Yes
Read DATA1 = 0 on each PD controller (1)?
No
Issue 'PBMe' 4CC task to each PD controller(1).
Yes
Read CMD1=0 on each PD controller (1)?
No
No
Yes
Read CMD1='!CMD' on any PD controller (1)?
Delay 10ms
Read CMD1 register
Write CMD1='PBMc' on I2C_EC bus (1)
Delay at least 500us
Yes
Entire Patch Bundle has been written?
No
Yes
The I2C transaction is Acknowledged?
No
Write Patch Bundle burst data on the I2C_EC bus using the slave address specified in DATA1.SlaveAddress. Terminate burst data with a Stop bit.

[1] Use the fundamental I2C slave address of each PD controller.

[2] This delay before reading the MODE register, is optional but recommended.

Figure 5-1. Flow for pushing a Patch Bundle over the I2C_EC bus to multiple PD controllers at the same time.

While the host is writing the Patch Bundle burst data, the I2C protocol in the following figure must be followed. The host may send the entire Patch Bundle in a single I2C transaction, or it may break it up into multiple transactions. The PD controller increments the pointer into its patch memory space with each byte received on the Patch Slave address that was configured by DATA1.SlaveAddress as part of the 'PBMs' 4CC Task. The EC may re-issue a 'PBMs' 4CC Task or it may issue a 'PBMe' 4CC Task in order to reset the pointer.



1	7	1	1	8	1	8	1		8	1	1
S	Patch Slave Address	Wr	A	Data Byte1	A	Data Byte2	A	...	Data Byte M	A	P

1	7	1	1	8	1	8	1		8	1	1
S	Patch Slave Address	Wr	A	Data Byte M+1	A	Data Byte M+2	A	...	Data Byte N	A	P

Figure 5-2. Protocol of Patch Bundle burst data assuming it is broken into two transactions.

5.3 EEPROM boot flow

During the boot process, the PD controller will load the Patch Bundle from the external EEPROM connected to the I2C3m port as described below. Once the PD controller is in the APP mode, i.e. MODE register reads as 'APP ', the host may write to the EEPROM as described below in order to update the Patch Bundle used for booting the PD controller. During the process the previous Patch Bundle is kept intact so that the PD controller can boot from it if anything goes wrong writing the new Patch Bundle into the EEPROM.

The Patch Bundle contains a FW patch image in addition to a set of configurations that set the default value of the HI registers.

The EEPROM is divided into two regions to allow for it being updated without invalidating the previous Patch Bundle until the new Patch Bundle has been verified. The active region is the one containing the latest Patch Bundle.

5.3.1 Boot process

At boot, the PD controller will first read the Header_ID from the Low Region at the address LowRegionStart + LowAppConfigOffset. If any error occurs in reading the Low Region Header_ID, the PD controller will then read the Header_ID from the High Region at the address HighRegionStart + HighAppConfigOffset. If any error occurs in reading the High Region Header_ID, the PD controller will loop back and try the Low Region again. The PD controller will only make two attempts, after that it aborts the EEPROM loading process.

If the PD controller reads the correct Header_ID (it is expecting 0xACE0_0001) in the Low Region, then it will begin reading the Patch Bundle from the Low Region. *If there is a CRC error while reading the Patch Bundle, the PD controller does not attempt to read from the High Region*. If the PD controller reads the correct Header_ID in the High Region, then it will begin reading the Patch Bundle from the High Region. If there is a CRC error while reading the Patch Bundle, the PD controller will attempt to read from the Low Region. However, the PD controller does not make more than two attempts on any region.

Therefore, when updating one of the regions of the EEPROM it is critical to verify the new Patch Bundle in the region before pointing the Region Start to it.

If the EEPROM loading process is aborted, then the PD controller will update the BOOT_STATUS register accordingly and assert the INT_EVENTx.ReadyForPatch interrupt. It will then wait indefinitely for the host to load a patch over the I2C_EC slave port or to issue a 'GAID' to reboot the PD controller. This is also what happens when there is no EEPROM present.

Figure 5-3 shows the memory map of the EEPROM and where the pointers and offsets reside assuming that the EEPROM has initially been written with the same Patch Bundle in both regions. The PD controller looks for the Header_ID of the Low Region at address LowRegionStart + LowAppConfigOffset, and it looks for the Header_ID of the High Region at the address HighRegionStart + HighAppConfigOffset.

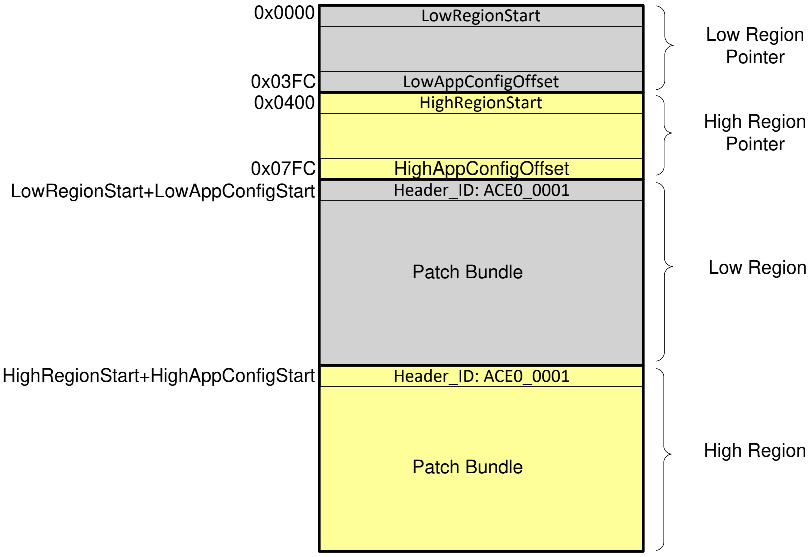


Figure 5-3. EEPROM memory map.

5.3.2 Updating the EEPROM image

When the host needs to update the Patch Bundle used for booting it should follow the process described in this section. The top-level flow is to update the region that the PD controller did NOT boot from as illustrated in Figure 5-4. The top-level flow executes the function UpdateRegionOfEeprom() that is illustrated in Figure 5-5.

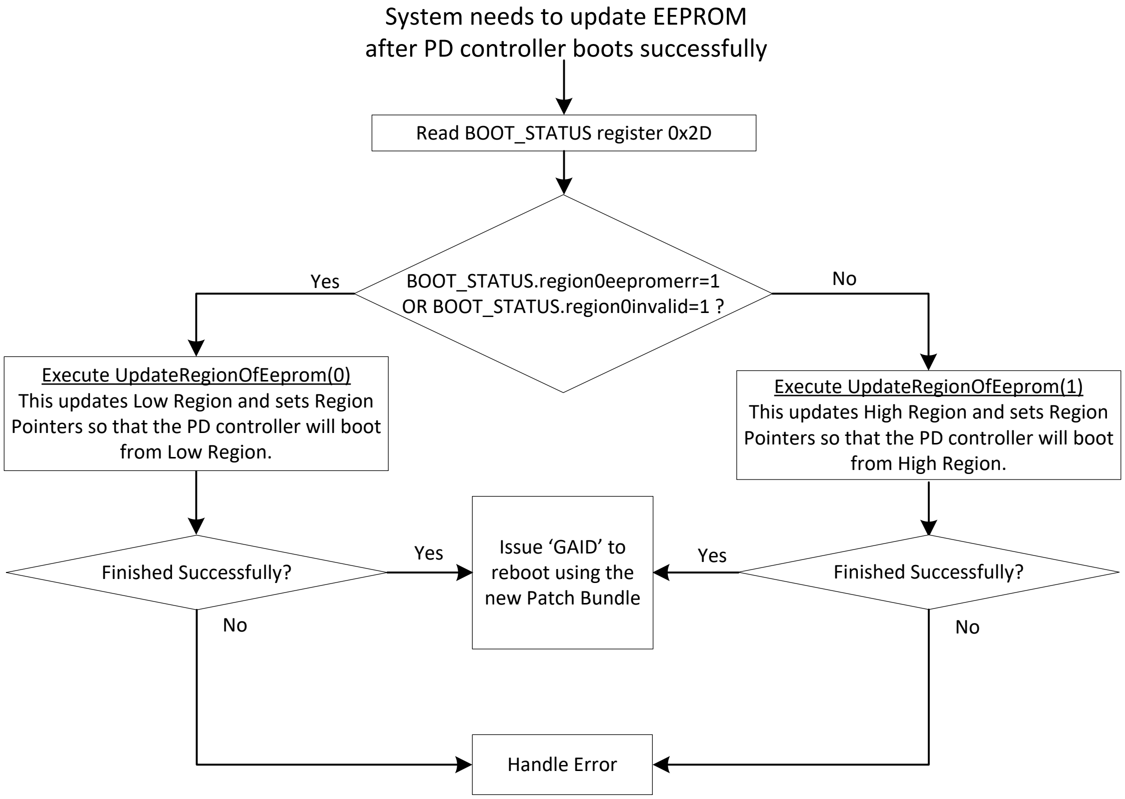

System needs to update EEPROM after PD controller boots successfully
Read BOOT_STATUS register 0x2D
BOOT_STATUS.region0eepromerr=1 OR BOOT_STATUS.region0invalid=1 ?
Yes
No
Execute UpdateRegionOfEeprom(0)
This updates Low Region and sets Region Pointers so that the PD controller will boot from Low Region.
Execute UpdateRegionOfEeprom(1)
This updates High Region and sets Region Pointers so that the PD controller will boot from High Region.
Finished Successfully?
Yes
Issue 'GAID' to reboot using the new Patch Bundle
Yes
Finished Successfully?
No
No
Handle Error

Figure 5-4. Flow for updating the EEPROM.

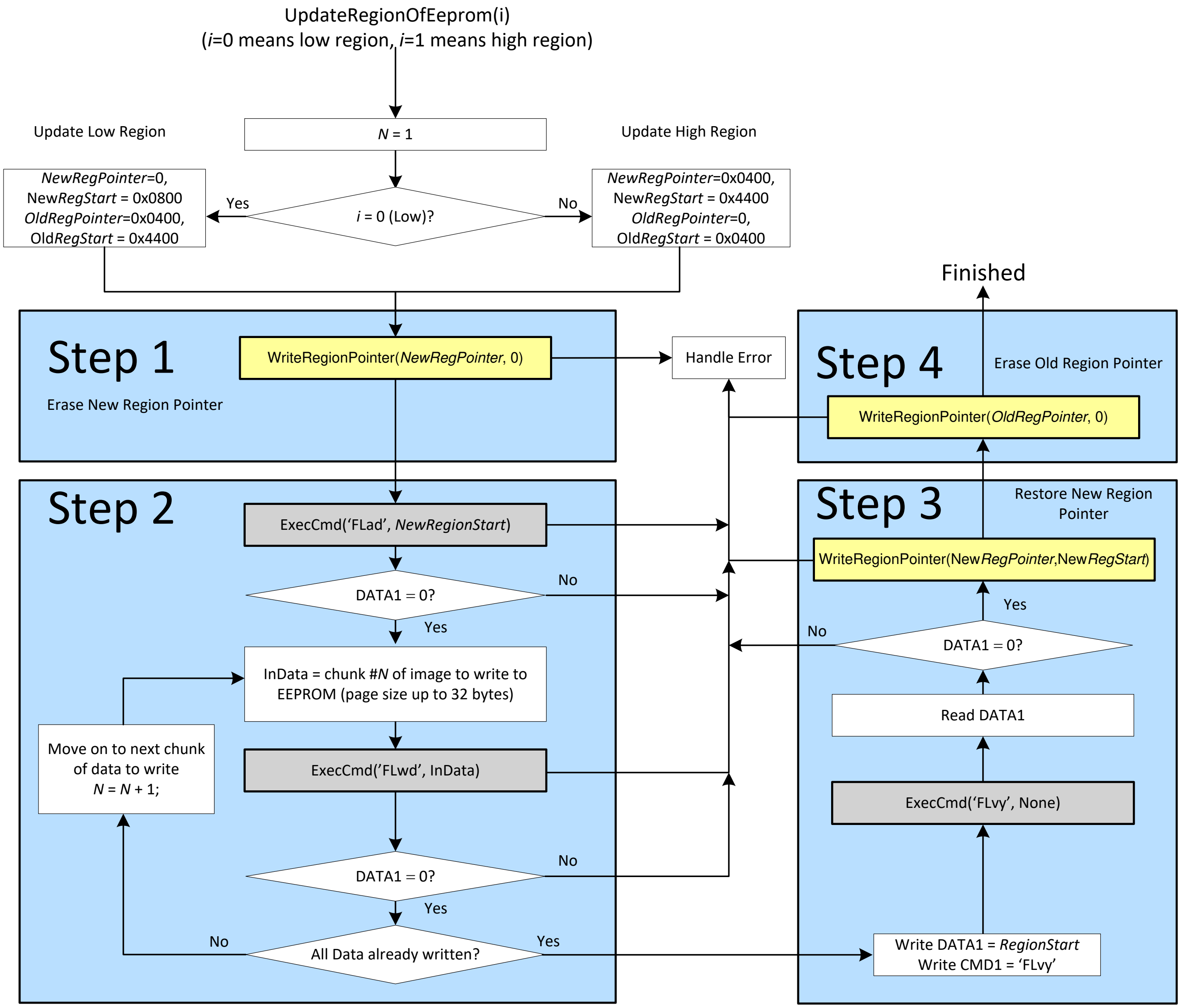

UpdateRegionOfEeprom(i)
(i=0 means low region, i=1 means high region)
N = 1
Update Low Region
Update High Region
i = 0 (Low)?
Yes
No
NewRegPointer=0, NewRegStart = 0x0800 OldRegPointer=0x0400, OldRegStart = 0x4400
NewRegPointer=0x0400, NewRegStart = 0x4400 OldRegPointer=0, OldRegStart = 0x0400
Step 1
Erase New Region Pointer
WriteRegionPointer(NewRegPointer, 0)
Handle Error
Finished
Step 4
Erase Old Region Pointer
WriteRegionPointer(OldRegPointer, 0)
Step 2
ExecCmd('FLad', NewRegionStart)
DATA1 = 0?
No
Yes
InData = chunk #N of image to write to EEPROM (page size up to 32 bytes)
Move on to next chunk of data to write N = N + 1;
ExecCmd('FLwd', InData)
DATA1 = 0?
No
Yes
All Data already written?
No
Yes
Step 3
Restore New Region Pointer
WriteRegionPointer(NewRegPointer,NewRegStart)
Yes
DATA1 = 0?
No
Read DATA1
ExecCmd('FLvy', None)
Write DATA1 = RegionStart Write CMD1 = 'FLvy'

Figure 5-5. Details of the UpdateRegionOfEeprom() function used to update EEPROM.

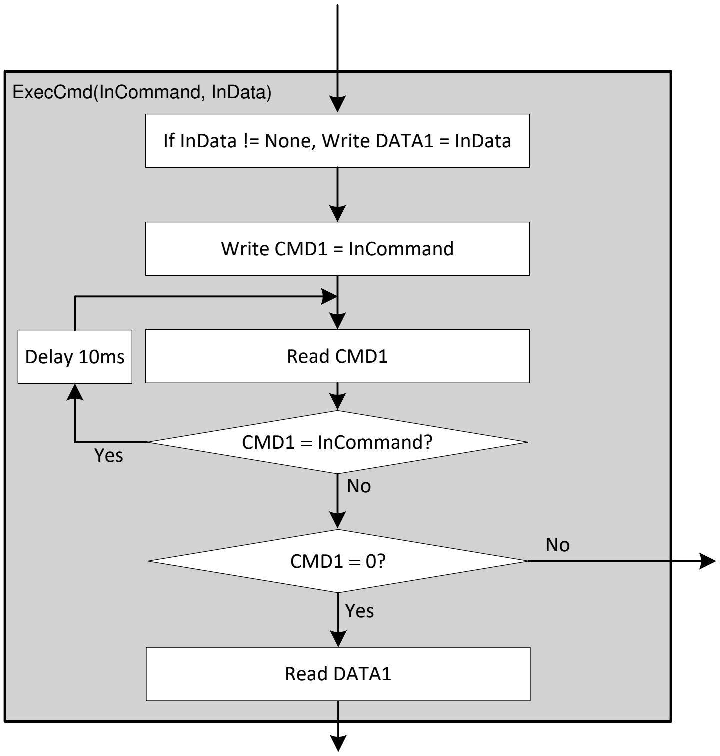


Figure 5-6. Details of the ExecCmd() block.

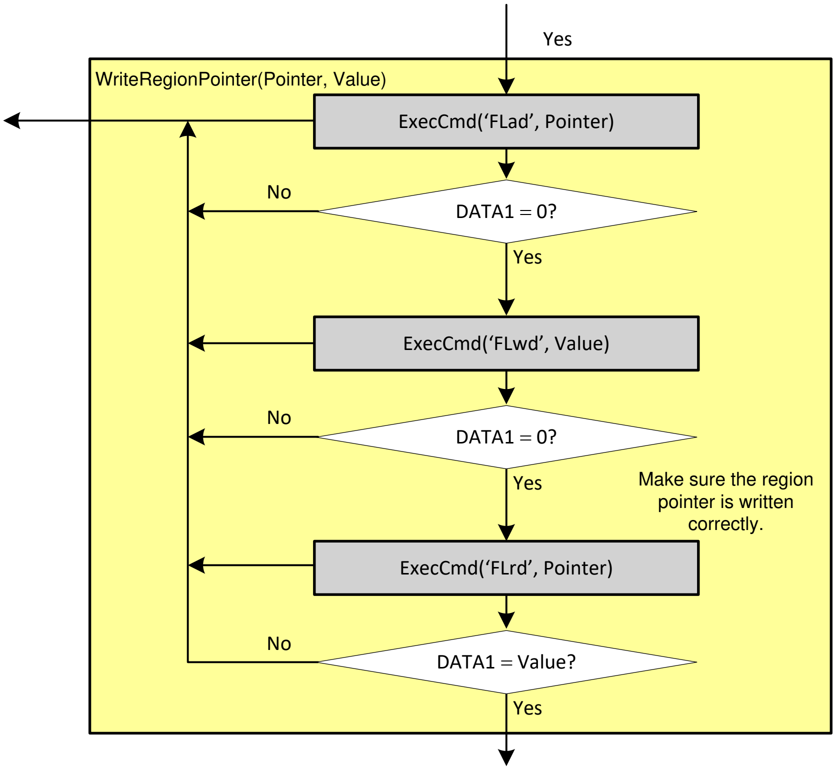


Figure 5-7. Details of the WriteRegionPointer() block.

5.3.3 EEPROM Update Example

In the example below, it is assumed that the initial state of the EEPROM is that the Low Region and the High Region both have the correct Patch Bundle. Figure 5-8 shows the EEPROM memory map for this initial condition.

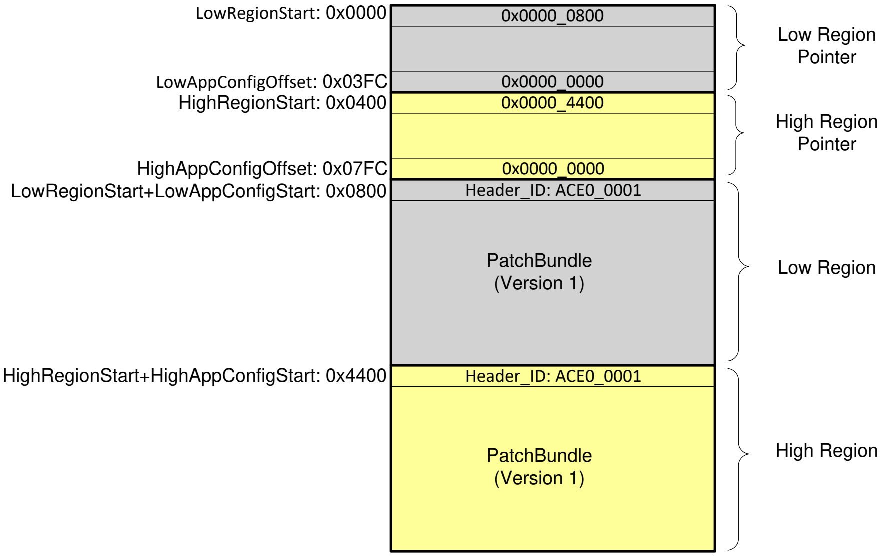


Figure 5-8. Initial state of the EEPROM. Both Low and High regions contain the same Patch Bundle. The PD controller will boot from the Low Region.

When the host needs to update the Patch Bundle that the PD controller uses for booting it first erases the High Region pointer so that if there is an interruption while writing the new Patch Bundle to the High Region it is guaranteed the PD controller won't try to load it. Specifically, Step 1 of the function UpdateRegionOfEeprom(1) from Figure 5-5 is executed, and Figure 5-9 shows the memory map after the High Region pointer has been erased successfully.

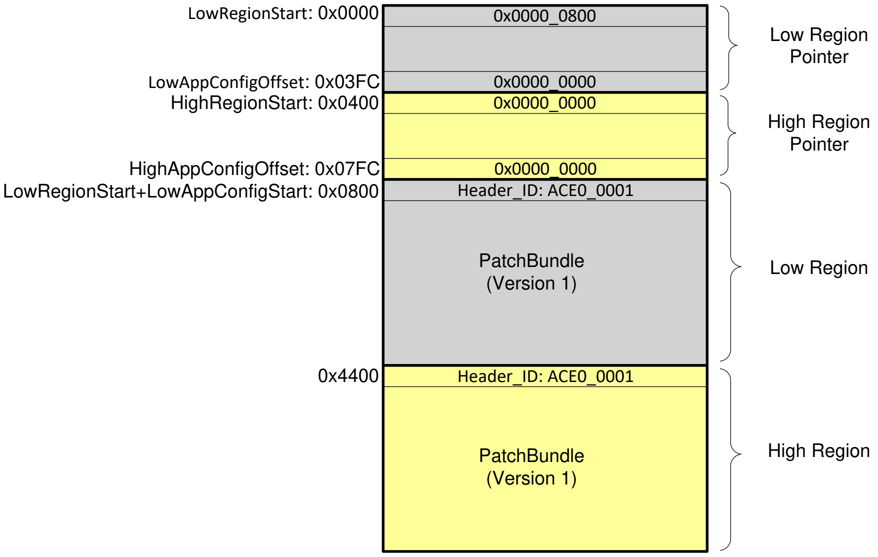


Figure 5-9. State of the EEPROM following UpdateRegionOfEeprom(1) Step 1. If booted in this state, Patch Bundle (Version 1) is loaded from Low Region.

Next, the host writes the new Patch Bundle (Version 2) to the High Region. Specifically, Step 2 of the function UpdateRegionOfEeprom(1) from Figure 5-5 is executed, and Figure 5-10 shows the memory map following this step. Note that if the PD controller boots with the EEPROM in this state, it will still load Patch Bundle (Version 1) from the Low Region.

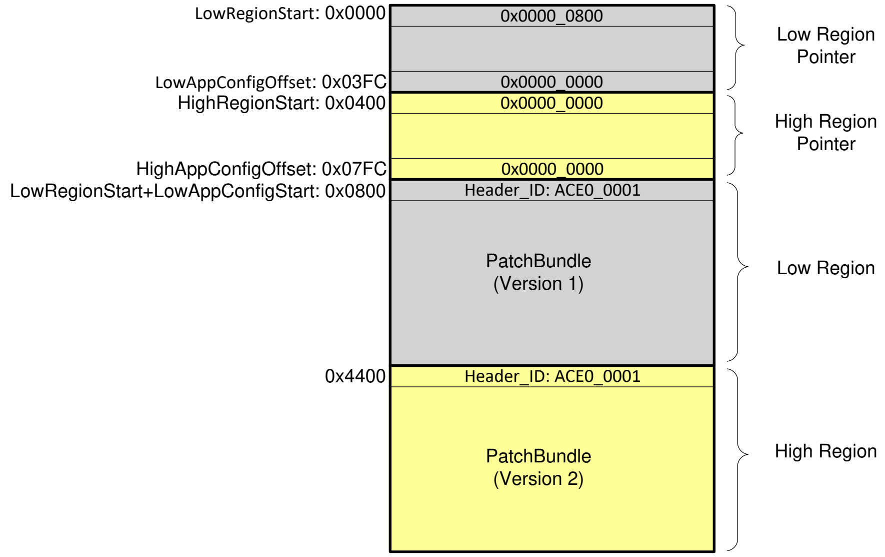


Figure 5-10. State of the EEPROM following UpdateRegionOfEeprom(1) Step 2. If booted in this state, Patch Bundle (Version 1) is loaded from Low Region.

Next, the host will verify the contents of the EEPROM High Region using the 'FLvy' command. If that succeeds, then the host will write 0x4400 into HighRegionStart at address 0x0400. This happens in Step 3 of UpdateRegionOfEeprom(1) in Figure 5-5, and Figure 5-11 shows the memory map after this is done. If the PD controller reboots with the EEPROM in this state, it will still first attempt to boot from the Low Region.

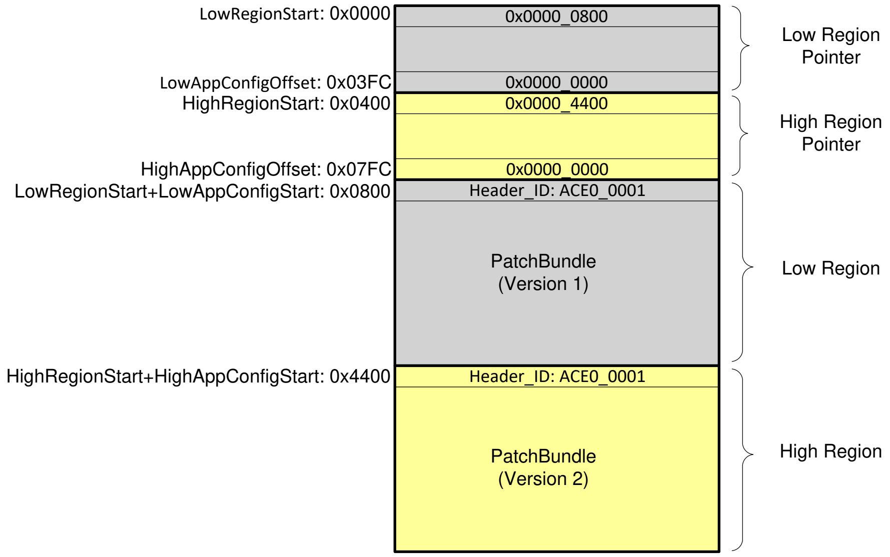


Figure 5-11. State of EEPROM following UpdateRegionOfEeprom(1) Step 3. If booted in this state, Patch Bundle (Version 1) is loaded from Low Region.

The last step is to erase the LowRegionStart value so that the PD controller will boot from the High Region. The host may use the WriteRegionPointer() functionality as shown in Step 4 of UpdateRegionOfEeprom(1) as illustrated in Figure 5-5. Figure 5-12 shows the memory map after Step 4 is complete. Since the LowRegionStart is now 0, the contents of the Low Region have no impact on how the PD controller boots.

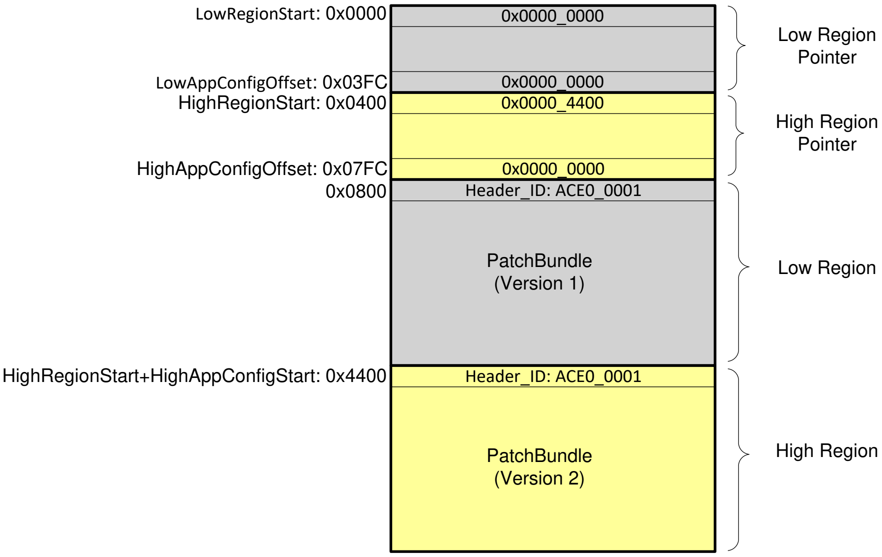


Figure 5-12. State of EEPROM following UpdateRegionOfEeprom(1) Step 4. If booted in this state, Patch Bundle (Version 2) is loaded from High Region.

The next time the host needs to update the EEPROM image it would execute UpdateRegionOfEeprom(0), and the process would proceed in a similar manner. The host may optionally execute UpdateRegionOfEeprom(0) immediately with the same new Patch Bundle it has just written to the High Region.

5.3.4 Source code example

This section gives example source code to implement the EEPROM flow described above.

5.3.4.1 UpdateRegionOfEeprom()

```
const uint32_t region_ptr_start[NUM_OF_REGIONS]            = {0x0  , 0x400 };
const uint32_t region_ptr_appconfig_offset[NUM_OF_REGIONS]  = {0x3FC, 0x7FC };
const uint32_t region_addr_patchbundle[NUM_OF_REGIONS]      = {0x800, 0x4400};

static int32_t UpdateRegionOfEeprom()
{
    s_AppContext *const pCtx = &gAppCtx;
    int32_t retVal = -1;

    UART_PRINT("\n\rActive Region is [%d] - Region being updated is [%d]\n\r",\
                                    pCtx->active_region, pCtx->inactive_region);

    /*
     * Region-0/Region-1 is currently active, hence update Region-1/Region-0 respectively
     */
    retVal = UpdateRegionOfEeprom_Step1(pCtx->inactive_region);
    if(0 != retVal)
    {
        UART_PRINT("Region[%d] update Step 1 failed.! Next boot will happen from Region[%d]\n\r",\
                                         pCtx->inactive_region, pCtx->active_region);
        goto error;
    }

    /*
     * Region-0/Region-1 is currently active, hence update Region-1/Region-0 respectively
     */
    retVal = UpdateRegionOfEeprom_Step2(pCtx->inactive_region);
    if(0 != retVal)
    {
        UART_PRINT("Region[%d] update Step 2 failed.! Next boot will happen from Region[%d]\n\r",\
                                         pCtx->inactive_region, pCtx->active_region);
        goto error;
    }

    /*
     * Write is through. Now verify if the content/copy is valid.
               * Update the corresponding region-pointer point to the new region.
     */
    retVal = UpdateRegionOfEeprom_Step3(pCtx->inactive_region);

    if(0 != retVal)
    {
        UART_PRINT("Region[%d] update Step 3 failed.! Next boot will happen from Region[%d]\n\r",\
                                         pCtx->inactive_region, pCtx->active_region);
        goto error;
    }

    /*
     * Invalidate the region-pointer of the old region.
     */
    retVal = UpdateRegionOfEeprom_Step4(pCtx->active_region);
    if(0 != retVal) {goto error;}

error:
    SignalEvent(APP_EVENT_END_UPDATE);
    return retVal;
}
```

5.3.4.2 UpdateRegionOfEeprom_Step1

```
static int32_t UpdateRegionOfEeprom_Step1(uint8_t region_number)
{
    int32_t     retVal         = -1;

    /*
```

```
     * First erases the region-pointer so that if there is an interruption while writing
     * the new Patch Bundle, it is guaranteed the PD controller won't try to load it.
     */
    retVal = WriteRegionPointer(region_number, 0);
    RETURN_ON_ERROR(retVal);

error:
    return retVal;
}
```

5.3.4.3 UpdateRegionOfEeprom_Step2()

```
static int32_t UpdateRegionOfEeprom_Step2(uint8_t region_number)
{
    uint8_t     outdata[MAX_BUF_BSIZE]  = {0};

    s_TPS_flad  fladInData      = {0};

    uint32_t    bytesUpdated    = 0;
    int32_t     retVal          = -1;
    int32_t     idx             = -1;

    /*
     * Set the start address for the next write
     */
    fladInData.flashaddr = region_addr_patchbundle[region_number];
    retVal = ExecCmd(FLad, sizeof(fladInData), (uint8_t *)&fladInData,
                  TASK_RET_CODE_LEN, &outdata[0]);
    RETURN_ON_ERROR(retVal);
    if(0 != outdata[1]) {retVal = -1; goto error;}

    for (idx = 0; idx < gSizeLowregionArray/PATCH_BUNDLE_SIZE; idx++)
    {
        /*
         * Execute FLwd with PATCH_BUNDLE_SIZE bytes of patch-data
         * in each iteration
         */
        retVal = ExecCmd(FLwd, PATCH_BUNDLE_SIZE,\
                      (uint8_t *)&tps6598x_lowregion_array[idx * PATCH_BUNDLE_SIZE],
                      TASK_RET_CODE_LEN, &outdata[0]);
        RETURN_ON_ERROR(retVal);
        if(0 != outdata[1]) {retVal = -1; goto error;}

        bytesUpdated += PATCH_BUNDLE_SIZE;
        Board_IF_Delay(75); /* in uSecs */
    }

    /* Push more bytes if any */
    if(gSizeLowregionArray > bytesUpdated)
    {
        retVal = ExecCmd(FLwd, gSizeLowregionArray - bytesUpdated,\
                      (uint8_t *)&tps6598x_lowregion_array[idx * PATCH_BUNDLE_SIZE],
                      TASK_RET_CODE_LEN, &outdata[0]);
        RETURN_ON_ERROR(retVal);
        if(0 != outdata[1]) {retVal = -1; goto error;}
    }

  error:
    return retVal;
}
```

5.3.4.4 UpdatingRegionOfEeprom_Step3()

```
static int32_t UpdateRegionOfEeprom_Step3(uint8_t new_region_number)
{
    uint8_t     outdata[MAX_BUF_BSIZE]  = {0};

    s_TPS_flvy  flvyInData      = {0};

    int32_t     retVal          = -1;

    flvyInData.flashaddr = region_addr_patchbundle[new_region_number];
    retVal = ExecCmd(FLvy, sizeof(flvyInData), (uint8_t *)&flvyInData, \
                    TASK_RET_CODE_LEN, &outdata[0]);
    if(0 != outdata[1]) {retVal = -1; goto error;}

    retVal = WriteRegionPointer(new_region_number, region_addr_patchbundle[new_region_number]);
    RETURN_ON_ERROR(retVal);

error:
    return retVal;
}
```

5.3.4.5 UpdatingRegionOfEeprom_Step4()

```
static int32_t UpdateRegionOfEeprom_Step4(uint8_t old_region_number)
{
    int32_t     retVal          = -1;

    retVal = WriteRegionPointer(old_region_number, 0);
    RETURN_ON_ERROR(retVal);

error:
    return retVal;
}
```

5.3.4.6 WriteRegionPointer()

```
static int32_t WriteRegionPointer(const uint8_t region_number, const uint32_t value)
{
    uint8_t     outdata[MAX_BUF_BSIZE]  = {0};
    s_TPS_flad  fladInData  = {0};

    uint32_t    regionVal   = 0;

    int32_t     retVal      = -1;

    fladInData.flashaddr = region_ptr_start[region_number];
    retVal = ExecCmd(FLad, sizeof(fladInData), (uint8_t *)&fladInData, TASK_RET_CODE_LEN,
&outdata[0]);
    RETURN_ON_ERROR(retVal);
    if(0 != outdata[1]) {retVal = -1; goto error;}

    retVal = ExecCmd(FLwd, sizeof(uint32_t), (uint8_t *)&value, TASK_RET_CODE_LEN, &outdata[0]);
    RETURN_ON_ERROR(retVal);
    if(0 != outdata[1]) {retVal = -1; goto error;}

    retVal = ExecCmd(FLrd, sizeof(uint32_t), (uint8_t *)&region_ptr_start[region_number],
sizeof(s_TPS_flrdassert), &outdata[0]);
    RETURN_ON_ERROR(retVal);
    regionVal = (outdata[4] << 24) |  (outdata[3] << 16) | (outdata[2] << 8) |  (outdata[1] << 0);
    if(value != regionVal) {retVal = -1; goto error;}

error:
    return retVal;
}
```

5.3.5 Recovering from EEPROM failure

If the EEPROM loading terminates without a valid Patch Bundle, then the INT_EVENTx.ReadyForPatch interrupt gets asserted. The host should read the BOOT_STATUS register 0x2D to discover why booting from EEPROM failed. Then the host must force the PD controller into the APP mode by pushing a patch using the 'PBMx' commands (see Section 5.2 for details). This could be the full Patch Bundle that is normally in EEPROM. Once the PD controller is in the APP mode, the host can write to the EEPROM using the 'FLxx' commands and correct the problem. The figure below shows the recommended boot flow.

This boot flow requires the host to be able to correct the EEPROM. If the host requires the PD controller to enable the sink path before it can boot, then the appropriate dead-battery configuration must be selected by the ADCINx pins. In this case, the SafeMode dead-battery configuration would not be appropriate.

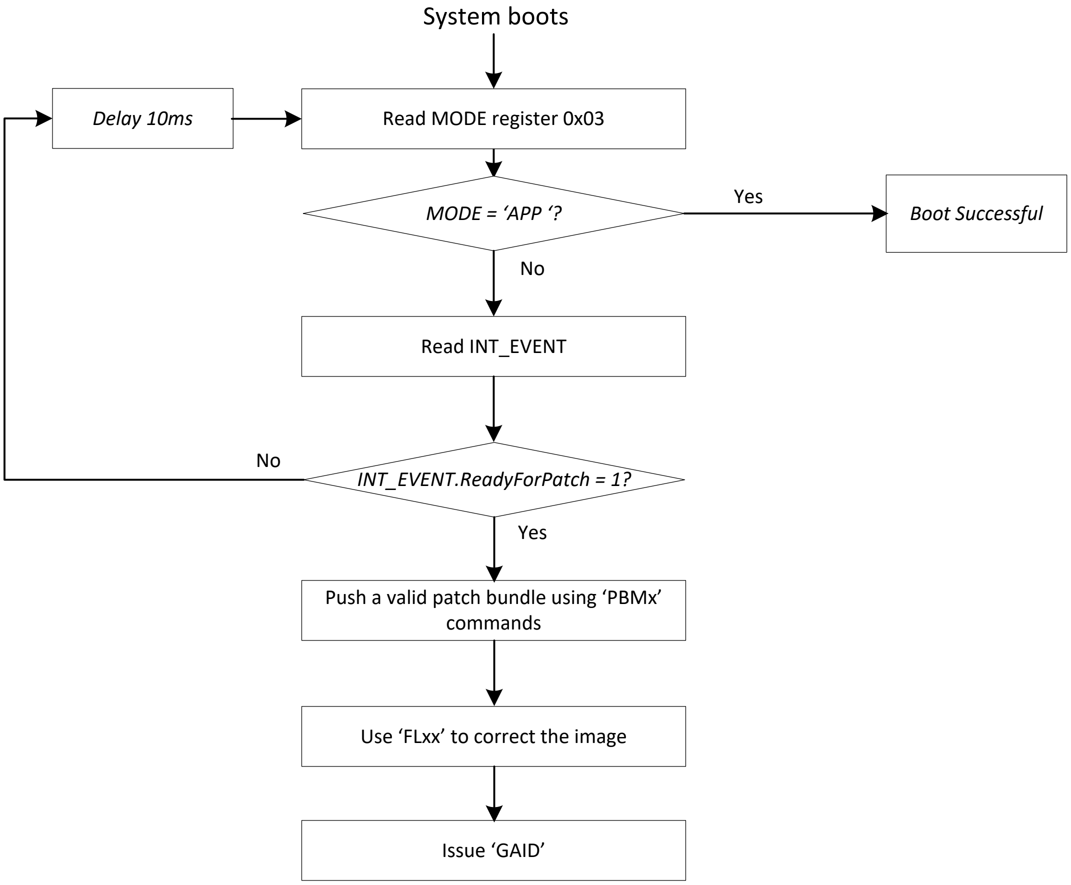


Figure 5-13. Recommended boot flow for EEPROM.

5.4 Fault Reporting

The PD controller reports various types of faults in different ways. This section summarizes the reporting.

Some registers are unique per port in a multi-port PD controller (See Table 1-1 for which registers have this property). In the following table, registers that are specific to PortA or PortB are noted as PortA.REGISTER or PortB.REGISTER.

Type of Fault	Reporting
Power path PP_5V1 goes into current limit.	• POWER_PATH_STATUS.PP1_Overcurrent = 1b • POWER_PATH_STATUS.PP1switch = 001b • PortA.INT_EVENTX.ErrorPowerEventOccurred = 1b • PortA.INT_EVENTX.Overcurrent = 1b • If GPIO Fault_Condition_Active_Low_ Event_Port1 is enabled, then that GPIO gets asserted. • PortA.DATA_STATUS.OvercurrentOrTemperature = 1b
Power path PP_5V2 goes into current limit.	• POWER_PATH_STATUS.PP2_Overcurrent = 1b • POWER_PATH_STATUS.PP2switch = 001b • PortB.INT_EVENTX.ErrorPowerEventOccurred = 1b • PortB.INT_EVENTX.Overcurrent = 1b • If GPIO Fault_Condition_Active_Low_ Event_Port2 is enabled, then that GPIO gets asserted. • DATA_STATUS.OvercurrentOrTemperature = 1b
Over-Voltage on PA_VBUS while PP_EXT1 enabled	• POWER_PATH_STATUS.PP3switch = 001b • PortA.INT_EVENTX.Error_PowerEventOccurred = 1b • PortA.PD_STATUS.ErrorRecoveryDetails = 4h
Over-Voltage on PB_VBUS while PP_EXT2 enabled	• POWER_PATH_STATUS.PP4switch = 001b • PortB.INT_EVENTX.Error_PowerEventOccurred = 1b • PortB.PD_STATUS.ErrorRecoveryDetails = 4h
Over-temperature on Port A (PP_5V1 or PP_CABLE1)	• POWER_PATH_STATUS.PP1overcurrent = 1b • POWER_PATH_STATUS.PP1switch = 001b • PortA.INT_EVENTX.ErrorPowerEventOccurred = 1b • PortA.INT_EVENTX.Overcurrent = 1b • If GPIO Event Fault_Condition_Active_Low_ Event_Port1 is enabled, then that GPIO gets asserted. • PortA.DATA_STATUS.OvercurrentOrTemperature = 1b • PortA.PD_STATUS.ErrorRecoveryDetails = 1h
Over-temperature on Port B (PP_5V1 or PP_CABLE1)	• POWER_PATH_STATUS.PP2overcurrent = 1b • POWER_PATH_STATUS.PP2switch = 001b • PortB.INT_EVENTX.ErrorPowerEventOccurred = 1b • PortB.INT_EVENTX.Overcurrent = 1b • If GPIO Event Fault_Condition_Active_Low_ Event_Port2 is enabled, then that GPIO gets asserted. • PortB.DATA_STATUS.OvercurrentOrTemperature = 1b • PortB.PD_STATUS.ErrorRecoveryDetails = 1h
PP_CABLE1 goes into current limit	• PortA.POWER_PATH_STATUS.PP1_CableOvercurrent = 1b • PortA.DATA_STATUS.OvercurrentOrOvertemperature = 1b • PortA.INT_EVENTX.Overcurrent = 1b
PP_CABLE2 goes into current limit	• PortB.POWER_PATH_STATUS.PP2_CableOvercurrent = 1b • PortB.DATA_STATUS.OvercurrentOrTemperature = 1b • PortB.INT_EVENTX.Overcurrent = 1b

5.5 Simple Power Management Feature

The PD controller has a simple power management (SPM) feature that is enabled by setting GLOBAL_SYSTEM_CONFIG.EnableSPM = 1b. This can automatically manage power within a dual-port controller and between two discrete PD controllers.

If GLOBAL_SYSTEM_CONFIG.EnableSPM = 1b when the PD controller receives a Request from the sink it does some additional processing to decide how to respond.

- If the other port within the PD controller has already negotiated a "high power" contract or if the Prevent_High_Current_Contract_Event GPIO is high, then the PD controller will reply with a Wait message. It will then transmit an updated Source Capabilities message that does not offer a high power contract.
  - In this context a "high power" contract is for any PDO index greater than 1 or PDO index equal to 1 with current greater than what is indicated in GLOBAL_SYSTEM_CONFIG.MinimumCurrentAdvertisement.
- Otherwise, the PD controller sends an Accept message and then drives the High_current_Contract_Active_Event output GPIO high, and then the other port will react depending upon its status (see table below).

When a "high power" port is detached or when the Prevent_High_Current_Contract_Event GPIO falls low, then the power sharing is re-evaluated and ports that have Explicit Contracts with the capability mismatch bit set get first priority. If the port partner has asserted the capability mismatch bit, the PD controller will immediately re-advertise TX_SOURCE_CAPS (i.e. offer the full power). Otherwise, the PD controller will use the TX_SOURCE_CAPS register the next time it transmits a Source Capabilities message in the normal course of its processing (e.g. if the sink sends a Get_Source_Cap message).

A common setting used to implement simple power management on TigerLake systems is listed below:

- GLOBAL_SYSTEM_CONFIG.EnableSPM = 1b
- GLOBAL_SYSTEM_CONFIG.MinimumCurrentAdvertisement = 1b
- PORT_CONTROL.TypeCCurrent = 0b

5.5.1 *SPM within a dual-port PD controller*

If GLOBAL_SYSTEM_CONFIG.EnableSPM = 1b when no port has a "high power" contract, the PD controller will advertise PORT_CONTROL.TypeCCurrent during implicit contracts and it will advertise TX_SOURCE_CAPS for explicit contracts. If another port already has a "high power" contract, the PD controller will limit the advertised current to GLOBAL_SYSTEM_CONFIG.MinimumCurrentAdvertisement for explicit contracts. Implicit contracts are always limited to 0.9A when another port has a "high power" contract.

Table 5-2. Simple power management within a dual-port controller.

GLOBAL_SYSTEM_CONFIG	PORT_CONTROL	PortX status	Port Y	
MinimumCurrentAdvertisement	TypeCCurrent		Implicit Contract	Explicit Contract
1.5A	0.9A	"high power": Explicit contract for 5V with > 1.5A.	0.9A	5V up to 1.5A
	1.5A		0.9A	
	3.0a		0.9A	
0.9A	0.9A	"high power": Explicit contract for 5V with > 0.9A.	0.9A	5V up to 0.9A
	1.5A		0.9A	
	3.0A		0.9A	
1.5A	0.9A	"low power": Explicit contract for 5V with ≤ 1.5A or an implicit contract.	0.9A	5V up to 3.0A
	1.5A		1.5A	
	3.0A		3.0A	
0.9A	0.9A	"low power": Explicit contract for 5V with ≤ 0.9A or an implicit contract.	0.9A	
	1.5A		1.5A	
	3.0A		3.0A	

5.5.2 SPM between two discrete PD controllers using GPIO

The GPIO Events Prevent_High_Current_Contract_Event and High_current_Contract_Active_Event may be used if this simple power management is for two discrete devices. See the following tables. The Prevent_High_Current_Contract_Event GPIO is connected to the High_current_Contract_Active_Event of the other device.

Table 5-3. Prevent_High_Current_Contract_Event Usage as input GPIO.

Edge Type	PD_STATUS	PD Contract Type	Action
	PresentPDRole		
falling	Source	Explicit	Re-transmit Source Capabilities as given by TX_SOURCE_CAPS.
falling	Source	Implicit	Set Type-C Rp to match PORT_CONTROL.TypeCCurrent.
rising	Source	Explicit	Re-transmit Source Capabilities with 5V max voltage and max current from GLOBAL_SYSTEM_CONFIG.MinimumCurrentAdvertisement.
rising	Source	Implicit	Set Type-C Rp to match GLOBAL_SYSTEM_CONFIG.MinimumCurrentAdvertisement.
N/A	Sink	N/A	None

Table 5-4. High_current_Contract_Active_Event GPIO control[1].

ACTIVE_CONTRACT_RDO		GLOBAL_SYSTEM_CONFIG	POWER_STATUS	Action
ObjectPosition	OperatingCurrentX	MinimumCurrentAdvertisement	PowerConnection	
> 1	> 900 mA	0b	1b	Drive GPIO high
>1	≤ 900 mA	0b	1b	
1	> 900 mA	0b	1b	
1	≤ 900 mA	0b	1b	Drive GPIO low
>1	> 1.5A	1b	1b	Drive GPIO high
>1	≤ 1.5A	1b	1b	
1	> 1.5A	1b	1b	
1	≤ 1.5A	1b	1b	Drive GPIO low
N/A	N/A	N/A	0b	

[1] This assumes that PD_STATUS.PresentPDRole=Source and GLOBAL_SYSTEM_CONFIG.EnableSPM=1b, otherwise the GPIO is driven low.

5.6 Alternate Mode Sequencing

The PD Controller always follows a set priority sequence for Alternate Mode entry. The sequence is listed below. If a given mode is disabled, then it is skipped. If a given mode is not possible due to incompatible Cable Plug or Port Partner, then it is skipped.

1. Enter USB for USB4 mode (if the DFP VDO in the TX_IDENTITY register has host capability set to USB4 host capable).
2. If USB4 mode was not entered, Thunderbolt 3 mode (if INTEL_VID_CONFIG.thunderBoltModeEnabled = 1 and thunderBoltAutoEntryAllowed = 1)
3. If neither USB4 mode nor TBT3 mode was entered, DisplayPort mode (if DP_SID_CONFIG.enableDPSID = 1 and DPModeAutoEntryAllowed = 1)
4. User SVID modes (if USER_VID_CONFIG.UserVidEnabled = 1 and UserModexAutoModeEntryAllowed = 1)
5. MIPI modes (if MIPI_VID_CONFIG.MipiVidEnabled = 1 and MipiDebugModeAutoEntryAllowed = 1)
6. If neither USB4 mode nor TBT3 mode nor DP mode was entered, Enter USB with USB3 mode (if the DFP VDO in the TX_IDENTITY register has host capability set to USB3.2 host capable).
7. If neither USB4 mode nor TBT3 mode nor DP mode nor USB3.2 mode was entered, Enter USB with USB2 mode (if the DFP VDO in the TX_IDENTITY register has host capability set to USB2 host capable).

Each port in a dual-port PD controller follows this sequence independently. There is not any port-to-port dependency automatically applied.

5.7 Simple UFP Data Role Management Feature

The PD controller has a simple UFP Data Role management feature that is enabled by setting GLOBAL_SYSTEM_CONFIG.EnableOneUFPPolicy = 0b. This can automatically manage the data-role within a dual-port controller and between two discrete PD controllers.

If GLOBAL_SYSTEM_CONFIG.EnableOneUFPPolicy = 0b when the PD controller with STATUS.DataRole = 1b (DFP) receives a DR_Swap from the Port Partner it does some additional processing to decide how to respond.

- If the other port within the PD controller already has STATUS.DataRole = 0b (UFP) or if the Prevent_DRSwap_To_UFP_Event GPIO is low, then the PD controller will reply with a Reject message.
- Otherwise, the PD controller sends an Accept message as long as PORT_CONTROL.ProcessSwapToDFP = 1b.
- The PD controller does not send a Wait message in response to a DR_Swap message.

Appendix A: Comparison of PD Controllers

A.1 TPS65994AD_F509.04.02 to TPS65994AC_F309.02.04 comparison

The baseline for comparison is TPS65994AC_F309.02.04.

Table A-1. Registers Added (+), Removed (-), or Unchanged (=) relative to baseline.

Register Name	TPS65994AD_F509.04.02
USB_CONFIG	+
USB_STATUS	+

Table A-2. Existing registers with fields Added (+), Removed (-), or Unchanged (=) relative to baseline.

Register.Field	TPS65994AD_F509.04.02
INT_EVENT1.DataResetStart	+
INT_EVENT1.AMDRenoirMuxError	+
INT_EVENT2.AMDRenoirMuxError	+
INT_MASK1.AMDRenoirMuxError	+
INT_MASK2.AMDRenoirMuxError	+
INT_CLEAR1.AMDRenoirMuxError	+
INT_CLEAR2.AMDRenoirMuxError	+
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_1_Port_1	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_2_Port_1	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_3_Port_1	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_1_Port_2	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_2_Port_2	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_3_Port_2	-
AUTO_NEGOTIATE_SINK.AutoDisableSinkUponCapMismatch	+
PD_STATUS.DataResetDetails	+
TX_IDENTITY.numValidVDOs	-
TX_IDENTITY.numValidVDOs	+
TX_IDENTITY.ProductTypeVdo0SOP	+
TX_IDENTITY.ProductTypeVdo1SOP	+
TX_IDENTITY.ProductTypeVdo2SOP	+
INTEL_VID_CONFIG.RetimerComplianceSupport	+
DATA_STATUS.ActiveCable	-
DATA_STATUS.RetimerOrRedriver	+
DATA_STATUS.DebugAlternateModeType	-
DATA_STATUS.ActiveCable	+
DATA_STATUS.USB4Connection	+
DATA_STATUS.PowerMismatch	-

Table A-3. Existing fields with functionality Added (+), Removed (-), or Unchanged (=) relative to baseline.

Register.Field	BitValue Affected	TPS65994AD_F509.04.02
PORT_CONFIG.TypeCSupportOptions	10b	-

Table A-4. GPIO Events Added (+), Removed (-), or Unchanged (=) relative to baseline.

GPIO Event Name	TPS65994AD_F509.04.02
PdNegotiationInProcess	+
AttachedAsSink	+
EnableSource_Port2	+
EnableSource_Port1	+

Table A-5. 4CC Tasks Added (+), Removed (-), or Unchanged (=) relative to baseline.

4CC	TPS65994AD_F509.04.02
Trig	+
DRST	+

A.2 TPS65993AD_F509.04.02 to TPS65993AC_F309.02.04 comparison

The baseline for comparison is TPS65993AC_F309.02.04.

Table A-6. Registers Added (+), Removed (-), or Unchanged (=) relative to baseline.

Register Name	TPS65993AD_F509.04.02
USB_CONFIG	+
USB_STATUS	+

Table A-7. Existing registers with fields Added (+), Removed (-), or Unchanged (=) relative to baseline.

Register.Field	TPS65993AD_F509.04.02
INT_EVENT1.DataResetStart	+
INT_EVENT1.AMDRenoirMuxError	+
INT_EVENT2.AMDRenoirMuxError	+
INT_MASK1.AMDRenoirMuxError	+
INT_MASK2.AMDRenoirMuxError	+
INT_CLEAR1.AMDRenoirMuxError	+
INT_CLEAR2.AMDRenoirMuxError	+
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_1_Port_1	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_2_Port_1	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_3_Port_1	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_1_Port_2	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_2_Port_2	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_3_Port_2	-
AUTO_NEGOTIATE_SINK.AutoDisableSinkUponCapMismatch	+
TX_IDENTITY.numValidVDOs	-
TX_IDENTITY.numValidVDOs	+
TX_IDENTITY.ProductTypeVdo0SOP	+

Table A-7. Existing registers with fields Added (+), Removed (-), or Unchanged (=) relative to baseline. (continued)

Register.Field	TPS65993AD_F509.04.02
TX_IDENTITY.ProductTypeVdo1SOP	+
TX_IDENTITY.ProductTypeVdo2SOP	+
INTEL_VID_CONFIG.RetimerComplianceSupport	+
DATA_STATUS.ActiveCable	-
DATA_STATUS.RetimerOrRedriver	+
DATA_STATUS.DebugAlternateModeType	-
DATA_STATUS.ActiveCable	+
DATA_STATUS.USB4Connection	+
DATA_STATUS.PowerMismatch	-
DATA_STATUS.PowerMismatch	-

Table A-8. Existing fields with functionality Added (+), Removed (-), or Unchanged (=) relative to baseline.

Register.Field	BitValue Affected	TPS65993AD_F509.04.02
PORT_CONFIG.TypeCSupportOptions	10b	-

Table A-9. GPIO Events Added (+), Removed (-), or Unchanged (=) relative to baseline.

GPIO Event Name	TPS65993AD_F509.04.02
PdNegotiationInProcess	+
AttachedAsSink	+
EnableSource_Port2	+
EnableSource_Port1	+

Table A-10. 4CC Tasks Added (+), Removed (-), or Unchanged (=) relative to baseline.

4CC	TPS65993AD_F509.04.02
Trig	+
DRST	+

A.3 TPS65992xAD_F509.05.02 to TPS65992xAC_F409.03.02 comparison

The baseline for comparison is TPS65992xAC_F409.03.02.

Table A-11. Registers Added (+), Removed (-), or Unchanged (=) relative to baseline.

Register Name	TPS65992xAD_F509.05.02
USB_CONFIG	+
USB_STATUS	+

Table A-12. Existing registers with fields Added (+), Removed (-), or Unchanged (=) relative to baseline.

Register.Field	TPS65992xAD_F509.05.02
INT_EVENT1.DataResetStart	+
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_1_Port_1	-

Table A-12. Existing registers with fields Added (+), Removed (-), or Unchanged (=) relative to baseline. (continued)

Register.Field	TPS65992xAD_F509.05.02
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_2_Port_1	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_3_Port_1	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_1_Port_2	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_2_Port_2	-
SET_Sx_APP_CONFIG.Trigger_App_Config_Group_3_Port_2	-
PD_STATUS.DataResetDetails	+
TX_IDENTITY.numValidVDOs	-
TX_IDENTITY.numValidVDOs	+
TX_IDENTITY.ProductTypeVdo0SOP	+
TX_IDENTITY.ProductTypeVdo1SOP	+
TX_IDENTITY.ProductTypeVdo2SOP	+
DATA_CONTROL.DpHostConnected	+
DATA_CONTROL.UsbHostConnected	+
INTEL_VID_CONFIG.RetimerComplianceSupport	+
DATA_STATUS.ActiveCable	-
DATA_STATUS.RetimerOrRedriver	+
DATA_STATUS.DebugAlternateModeType	-
DATA_STATUS.ActiveCable	+
DATA_STATUS.USB4Connection	+
I2CMASTER_CONFIG.TimePeriod	+
I2CMASTER_CONFIG.HSModeEnable	+
I2CMASTER_CONFIG.SClkLowTime	+
I2CMASTER_CONFIG.SClkHighTime	+
DATA_STATUS.PowerMismatch	-

Table A-13. Existing fields with functionality Added (+), Removed (-), or Unchanged (=) relative to baseline.

Register.Field	BitValue Affected	TPS65992xAD_F509.05.02
PORT_CONFIG.AMDI2CMuxEnable	0b	+
PORT_CONFIG.AMDI2CMuxEnable	1b	+

Table A-14. GPIO Events Added (+), Removed (-), or Unchanged (=) relative to baseline.

GPIO Event Name	TPS65992xAD_F509.05.02
PdNegotiationInProcess	+
AttachedAsSink	+
EnableSource_Port2	+
EnableSource_Port1	+

Table A-15. 4CC Tasks Added (+), Removed (-), or Unchanged (=) relative to baseline.

4CC	TPS65992xAD_F509.05.02
Trig	+
DRST	+

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265