TPS25730 Technical Reference Manual

Technical Reference Manual

Texas Instruments

Literature Number: SLVUCJ7
OCTOBER 2023

Table of Contents

List of Figures

List of Tables

This page intentionally left blank.

Preface
Read This First

About This Manual

This manual covers the features and peripherals supported by the TPS25730 USB Type-C® and PD controller devices. The document covers the Host Interface (4CC) Command and register read/write descriptions.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers can be shown with the suffix h or the prefix 0x. For example, the following number is 40 hexadecimal (decimal 64): 40h or 0x40.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties with default reset value below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure can have one of multiple meanings:
    - Not implemented on the device
    - Reserved for future device expansion
    - Reserved for TI testing
    - Reserved configurations of the device that are not supported
  - Writing nondefault values to the Reserved bits could cause unexpected behavior and should be avoided.

Glossary

TI Glossary — This glossary lists and explains terms, acronyms, and definitions.

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

Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

Trademarks

TI E2E™ is a trademark of Texas Instruments.
USB Type-C® is a registered trademark of USB Implementers Forum.
All trademarks are the property of their respective owners.

This page intentionally left blank.

Chapter 1
Introduction

1.1 Introduction

1.1.1 Purpose and Scope

Note

This section is for advanced users and the features listed here are only optional. This document is not necessary for simple power applications. An EC or Host is required in your system to implement some of the features described in this document.

This document describes the Host Interface for the TPS25730 Type-C Port Switch / Power Delivery (PD) Controller device.

1.2 PD Controller Host Interface Description

1.2.1 Overview

The PD Controller provides one I2C target. The I2Ct target is meant to be connected to an Embedded Controller (EC).

The Host Interface defines how the registers are accessed from I2C target port ands and all target addresses. target Address #1 is selected by the customer using the ADCIN1 and ADCIN2 pins on the PD controller.

The Host Interface provides general status information to the controller 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. Chapter 2 provides detailed Unique Address Interface register descriptions.

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

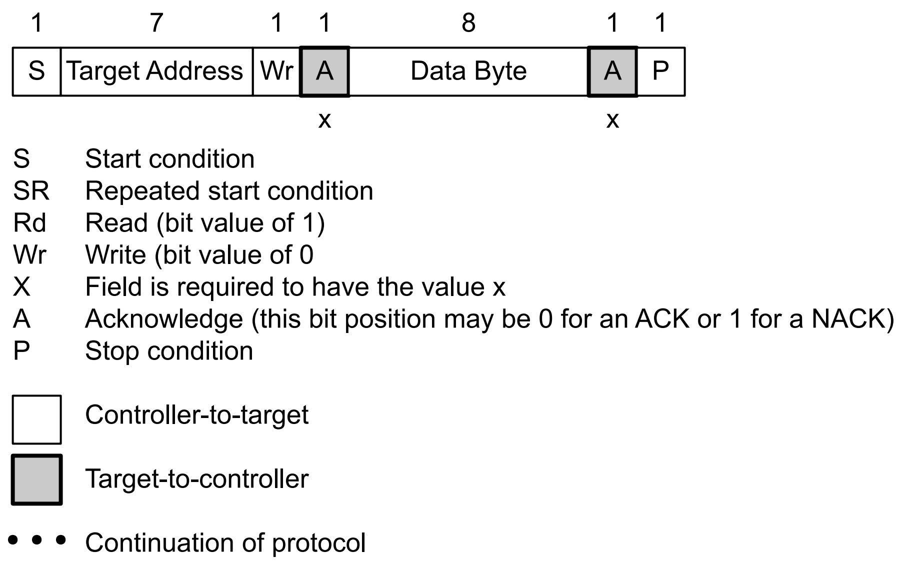

Figure 1-1. I2C Read/Write Protocol Key

1.3 Unique Address Interface

1.3.1 Unique Address Interface Protocol

The Unique Address Interface allows for complex interactions between an I2C controller and a single PD Controller. The I2C target unique address is used to receive or respond to Host Interface protocol commands. Figure 1-2 and Figure 1-3 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, controller may issue the stop bit without writing N bytes. Similarly, during a register read, the controller may issue the stop bit before reading all N bytes.

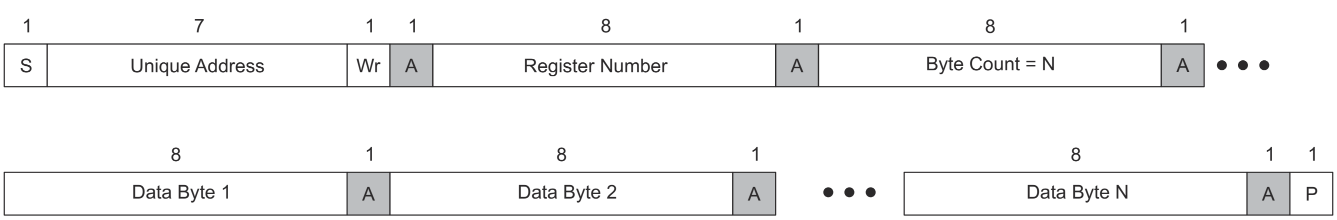

Figure 1-2. I2C Unique Address Write Register Protocol

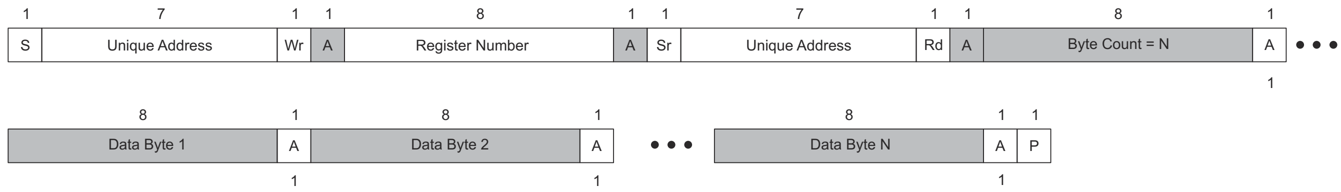

Figure 1-3. I2C Unique Address Read Register Protocol

This page intentionally left blank.

Chapter 2
Unique Address Interface Register Detailed Descriptions

2.1 0x31 RX_SINK_CAPS Register

Table 2-1. 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 2-2. 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).

2.2 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 must send a *Sink Capabilities* message.

Table 2-3. 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 2-4. 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 2-6.
Bytes 22-25: PDO #6 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO6	Sixth Sink Capabilities PDO contents. See Table 2-6.
Bytes 18-21: PDO #5 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO5	Fifth Sink Capabilities PDO contents. See Table 2-6.
Bytes 14-17: PDO #4 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO4	Fourth Sink Capabilities PDO contents. See Table 2-6.
Bytes 10-13: PDO #3 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO3	Third Sink Capabilities PDO contents. See Table 2-6.
Bytes 6-9: PDO #2 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO2	Second Sink Capabilities PDO contents. See Table 2-6.
Bytes 2-5: PDO #1 (treated as a 32-bit little endian value)		
31:0	TXSinkPDO1	First Sink Capabilities PDO contents. See Table 2-5.
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 2-5. 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
27:26	Reserved
25	Dual-Role Data, this is overridden by the logical OR of the ProcessSwapToUFP, ProcessSwapToDFP, InitiateSwapToUFP, and InitiateSwapToDFP fields in the PORT_CONTRL register.
24:20	Reserved
19:10	Voltage
9:0	Operational Current

Table 2-6. Other PDO's.

<table>
<tr><th rowspan="2">Bits(s)</th><th colspan="4">Description</th></tr>
<tr><th>Fixed Supply</th><th>Variable Supply</th><th>Battery Supply</th><th>APDO (PPS)</th></tr>
<tr><td>31:30</td><td>00b</td><td>01b</td><td>10b</td><td>11b</td></tr>
<tr><td>29:28</td><td rowspan="3">Reserved.</td><td rowspan="3">Maximum Voltage</td><td rowspan="3">Maximum Voltage</td><td>00b</td></tr>
<tr><td>27:25</td><td>Reserved</td></tr>
<tr><td>24:20</td><td rowspan="2">MaxPpsVoltage</td></tr>
<tr><td>19:17</td><td rowspan="3">Voltage</td><td rowspan="3">Minimum Voltage</td><td rowspan="3">Minimum Voltage</td></tr>
<tr><td>16</td><td>Reserved</td></tr>
<tr><td>15:10</td><td rowspan="2">MinPpsVoltage</td></tr>
<tr><td>9:8</td><td rowspan="3">Operational Current</td><td rowspan="3">Operational Current</td><td rowspan="3">Operational Power</td></tr>
<tr><td>7</td><td>Reserved</td></tr>
<tr><td>6:0</td><td>MaxPpsCurrent</td></tr>
</table>

2.3 0x34 ACTIVE_CONTRACT_PDO Register

Table 2-7. 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 2-8. 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, after it is Accepted by Source.

2.4 0x35 ACTIVE_CONTRACT_RDO Register

Table 2-9. 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 2-10. 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.

Chapter 3
4CC Task Detailed Descriptions

3.1 Overview

Note

This section is for advanced users and the features listed here are only optional. An EC or Host is required in your system to implement the features described in the following section.

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 DATA registers always ensure the proper output data is loaded into those registers before setting the CMD register to 0 to indicate Task completion. DATA is never modified by PD Controller after CMD 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 DATA register. The standard Task response byte is defined in Table 3-1. The remaining DATA bytes may be used at each Task's discretion.

Table 3-1. Standard Task Response

<table>
<tr><th>Description</th><th colspan="4">Tasks are a special form of Tasks that return a status code in the first byte of the DATA register.</th></tr>
<tr><td rowspan="10">Output DATA</td><td>Bit</td><td>Name</td><td colspan="2">Description</td></tr>
<tr><td colspan="4">Byte 1: Task Return Code</td></tr>
<tr><td>7:4</td><td>Reserved</td><td colspan="2">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.</td></tr>
<tr><td rowspan="7">3:0</td><td rowspan="7">TaskResult</td><td colspan="2">Standard Task return codes.</td></tr>
<tr><td>0x0</td><td>Task completed successfully.</td></tr>
<tr><td>0x1</td><td>Task timed-out or aborted by 'ABRT' Request.</td></tr>
<tr><td>0x2</td><td>Reserved.</td></tr>
<tr><td>0x3</td><td>Task rejected.</td></tr>
<tr><td>0x4</td><td>Task rejected because the Rx Buffer was locked. This is for Tasks that require the PD controller to use the Rx Buffer.</td></tr>
<tr><td>0x5-0xF</td><td>Reserved for standard Tasks. May be used by certain Tasks for Task-specific error codes. Treated as an error when encountered.</td></tr>
</table>

3.2 PD Message Tasks

3.2.1 'GSrC' - PD Get Source Capabilities

Table 3-2. '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 3-1.
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.

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

DATE	REVISION	NOTES
October 2023	*	Initial Release

This page intentionally left blank.

IMPORTANT NOTICE AND DISCLAIMER

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