Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

UPD350

Highly Integrated Small Form Factor USB Type-C™ Power Delivery 3.0 Port Controller

Highlights

- Companion PD Controller for Microchip USB Hub / SoC
- Small Form Factor QFN Package
- Integrated Analog Discrete Components Reduce Bill of Materials and Design Footprint
- USB Power Delivery 3.0 Compliant MAC
- USB Type-C [1] Connector Support with Connection Detection and Control
- I^2C/SPI [2] Interface for CPU/SoC Communication
- USB Type-C™ Alternate Mode Support
- Dual Role Port (DRP) and Fast Role Swap (FRS) Support with DRP offload mode

Target Applications

- Notebook Computers
- All-in-One/Desktop PCs
- Smartphones
- Tablets
- Monitors
- Docking Stations
- HDTVs
- Printers
- Automotive Breakout Boxes
- Multi-port Chargers

Key Benefits

- Integrated Analog Discrete Components
  - VCONN FETs with Rp/Rd Switching
  - Dead Battery Rd termination
  - Programmable Current Sense for Overcurrent Conditions
  - Voltage Sense for Overvoltage Conditions
- Integrated 3.3V Power Switch
  - Provides Dead Battery Support [2]
  - Automatically Switch between VBUS and Main +3.3V
- USB Power Delivery MAC
  - Compliant with USB Power Delivery Specification Revision 3.0
  - Power Delivery Packet Framing
  - CRC Checking/Generation
  - 4B/5B Encoding/Decoding
  - BMC Encoding/Decoding
  - EOP/SOP Generation for PD Frames
  - SOP Detection and SOP Header Processing
  - Separate RX/TX FIFOs
  - Automatic GoodCRC Message Generation
  - Automatic Retry Generation
  - Error Handling
  - Low Standby Power Support
- USB Type-C Cable Detect Logic
  - Auto Cable Attach & Orientation Detection
  - Routes Baseband Communication to Respective CC Pin per Detected Orientation
  - VCONN Supply Control for Active Cable
  - Configurable Downstream Facing Port (DFP) and Upstream Facing Port (UFP) Modes
  - Charging Current Capability Detection
  - Detection of Debug Accessory Mode, Audio Adapter Accessory Mode
- I^2C/SPI Interface Supports Communication/Configuration via Microchip USB Power Delivery hub or supported embedded controller [2]
- Alternate Mode Support
  - DisplayPort™ and other Major Protocols
- **CFG_SEL** Pin for Selection of Device Mode and I^2C addresses [2]
- Power and I/Os
  - Integrated 1.8V Voltage Regulator
  - 10 Configurable General Purpose I/O Pins
- Package
  - 28-QFN (4.0mm x 4.0mm)
- Environmental Product Options
  - Commercial (0°C to +70°C)
  - Industrial (-40°C to +85°C)
  - Automotive AEC-Q100 Grade 3 (-40°C to +85°C)

[1] *USB Type-C™ and USB-C™ are trademarks of USB Implementers Forum.*

[2] *Available only in select UPD350 configurations.*

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Documentation

To obtain the most up-to-date version of this documentation, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include -literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

Table of Contents

1.0 PREFACE

1.1 Glossary of Terms

TABLE 1-1: GLOSSARY OF TERMS

Term	Definition
ADC	Analog to Digital Converter
AFE	Analog Front End
BCI	Baseband CC Interface
Billboard	USB Billboard Device. A required USB device class for UFPs which support Alternate Modes in order to provide product information to the USB Host.
BIST	Built-In Self Test
BMC	Bi-phase Mark Coding
Byte	8-bits
CC	Generic reference to USB Type-C™ Cable / Connector CC1/CC2 pins
CSR	Control and Status Register
DB	Dead Battery
DFP	Downstream Facing Port (USB Type-C™ Specification definition)
DP	DisplayPort (a VESA standard interface)
DPM	Device Policy Manager (PD Specification definition)
DRP	Dual Role Power (USB Type-C™ Specification definition)
DWORD	32-bits
EC	Embedded Controller
EP	USB Endpoint
FIFO	First In First Out buffer
FW	Firmware
FS	Full-Speed
Host	External system (Includes processor, application software, etc.)
HPD	Hot-Plug Detect functionality as defined by DisplayPort and DisplayPort Alternate Mode specifications
HS	High-Speed
HW	Hardware (Refers to function implemented by the device)
IC	Integrated Circuit
IFC	InterFrame Gap
LDO	Linear Drop-Out regulator
MAC	Media Access Controller
Microchip	Microchip Technology Incorporated
N/A	Not Applicable
OCS	Over-Current Sense
PCS	Physical Coding Sublayer
PD / UPD	USB Power Delivery
PIO	General Purpose I/O
PMIC	Power Management Integrated Circuit
POR	Power-On Reset
PRBS	Pseudo Random Binary Sequence
QWORD	64-bits
SA	Source Address

TABLE 1-1: GLOSSARY OF TERMS (CONTINUED)

Term	Definition
SBU	SideBand Use
SCSR	System Control and Status Register
SPM	System Policy Manager (PD Specification definition)
SS	SuperSpeed
SVDM	Standard/Vendor Defined Message (PD Specification definition)
SVID	Standard/Vendor IDentity (PD Specification definition)
TCPC	USB Type-C Port Controller
UFP	Upstream Facing Port (USB Type-C™ Specification definition)
USB	Universal Serial Bus
USB Type-C	USB Type-C Cable / Connector
VDO	Vendor-defined Object (PD Specification definition)
VSM	Vendor Specific Messaging
WORD	16-bits
ZLP	Zero Length USB Packet

1.2 Buffer Types

TABLE 1-2: BUFFER TYPES

Buffer Type	Description
IS	Schmitt-triggered input
I2C	I^{2}C interface
O8	Output with 8 mA sink and 8 mA source
OD8	Open-drain output with 8 mA sink
PU	70k (typical) internal pull-up. Unless otherwise noted in the pin description, internal pull-ups are always enabled. **Note:** Internal pull-up resistors prevent unconnected inputs from floating. Do not rely on internal resistors to drive signals external to the device. When connected to a load that must be pulled high, an external resistor must be added.
AIO	Analog bidirectional
P	Power pin

Note: Digital signals are not 5V tolerant unless specified.

Note: Refer to Section 14.5, "DC Characteristics," on page 53 for the electrical characteristics of the various buffers.

1.3 Register Nomenclature

TABLE 1-3: REGISTER NOMENCLATURE

Register Bit Type Notation	Register Bit Description
R	**Read:** A register or bit with this attribute can be read.
W	**Write:** A register or bit with this attribute can be written.
RO	**Read only:** Read only. Writes have no effect.
RS	**Read to Set:** This bit is set on read.
WO	**Write only:** If a register or bit is write-only, reads will return unspecified data.
W1S	**Write One to Set:** Writing a one sets the value. Writing a zero has no effect.
W1C	**Write One to Clear:** Writing a one clears the value. Writing a zero has no effect.
WC	**Write Anything to Clear:** Writing anything clears the value.
LL	**Latch Low:** Clear on read of register.
LH	**Latch High:** Clear on read of register.
SC	**Self-Clearing:** Contents are self-cleared after the being set. Writes of zero have no effect. Contents can be read.
RO/LH	**Read Only, Latch High:** Bits with this attribute will stay high until the bit is read. After it is read, the bit will remain high, but will change to low if the condition that caused the bit to go high is removed. If the bit has not been read, the bit will remain high regardless of a change to the high condition.
NASR	**Not Affected by Software Reset.** The state of NASR bits do not change on assertion of a software reset.
RESERVED	**Reserved Field:** Reserved fields must be written with zeros, unless otherwise indicated, to ensure future compatibility. The value of reserved bits is not guaranteed on a read.

1.4 References

- NXP I^{2}C-Bus Specification (UM10204, April 4, 2014): www.nxp.com/documents/user_manual/UM10204.pdf
- USB Power Delivery and USB Type-C™ Specifications: http://www.usb.org/developers/docs/usb_31_102015.zip
- VESA DisplayPort Alternate Mode Specification 1.0: http://www.vesa.org

2.0 INTRODUCTION

2.1 General Description

The UPD350 is a highly integrated, small form factor USB Type-C™ Power Delivery (PD) Port Controller designed to adhere to the *USB Type-C™ Cable and Connector Specification* and *USB Power Delivery 3.0 Specification*. The UPD350 provides cable plug orientation and detection for a USB Type-C receptacle and implements baseband communication with a partner USB Type-C device via the integrated USB Power Delivery 3.0 MAC. The UPD350 is designed to function as a Companion Power Delivery Controller to an external Microchip MCU/SoC or USB hub using the integrated I^{2}C/SPI interface. The device is capable of controlling up to 100W of Power Delivery current and voltage using an external power device. Alternatively, the UPD350 can operate as a standalone UFP basic Type-C (non-PD) controller.

The UPD350 integrates many of the analog discrete components required for USB Type-C PD applications, including two VCONN FETs with Rp/Rd switching, a power switch, and current and voltage sense circuitry for over-voltage/current detection. By integrating many of the analog discrete components required for USB Type-C PD applications, the UPD350 provides a low cost, low power, small footprint solution for consumer (notebooks, desktop PCs, smartphones, tablets, monitors, docking stations) applications.

To enable the UPD350 to efficiently support dead battery use cases, an integrated power switch is provided to select between two external 3.3V supplies (VBUS and main). This effectively allows connection detection and system wakeup without external processor intervention (external processor in sleep mode).

The UPD350 is capable of negotiating alternate modes over USB Type-C connectors using the Power Delivery 3.0 protocol. DisplayPort operation over USB Type-C connectors is supported in addition to other major protocols.

A system diagram utilizing the UPD350 is shown in Figure 2-1. An internal block diagram of the UPD350 is shown in Figure 2-2.

FIGURE 2-1: SYSTEM BLOCK DIAGRAM

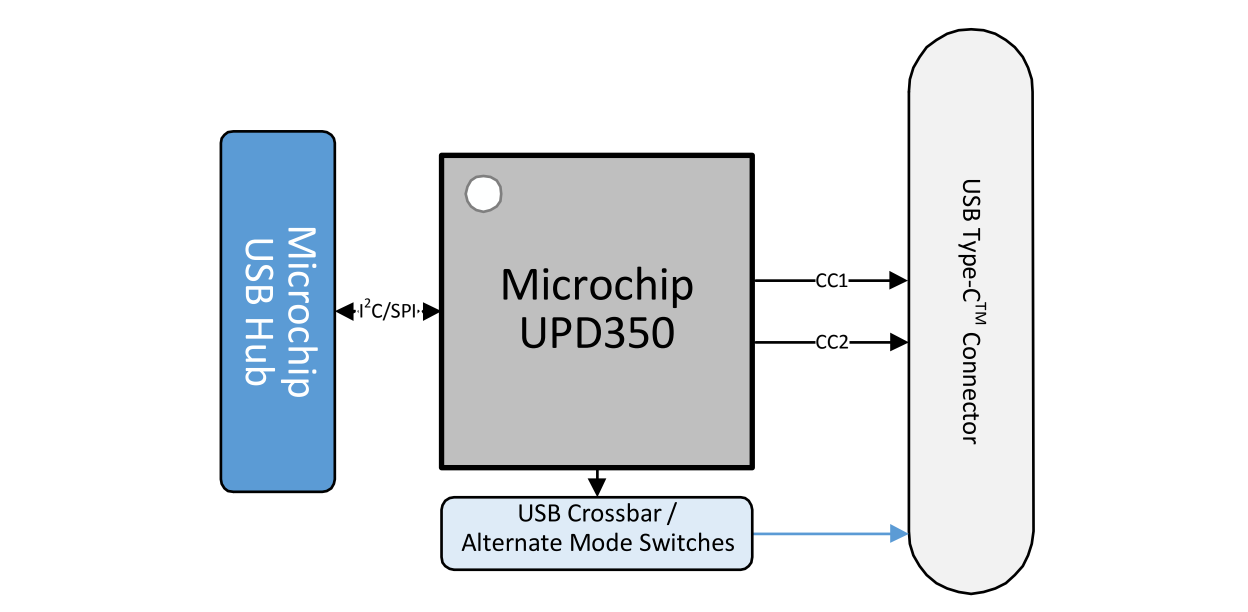
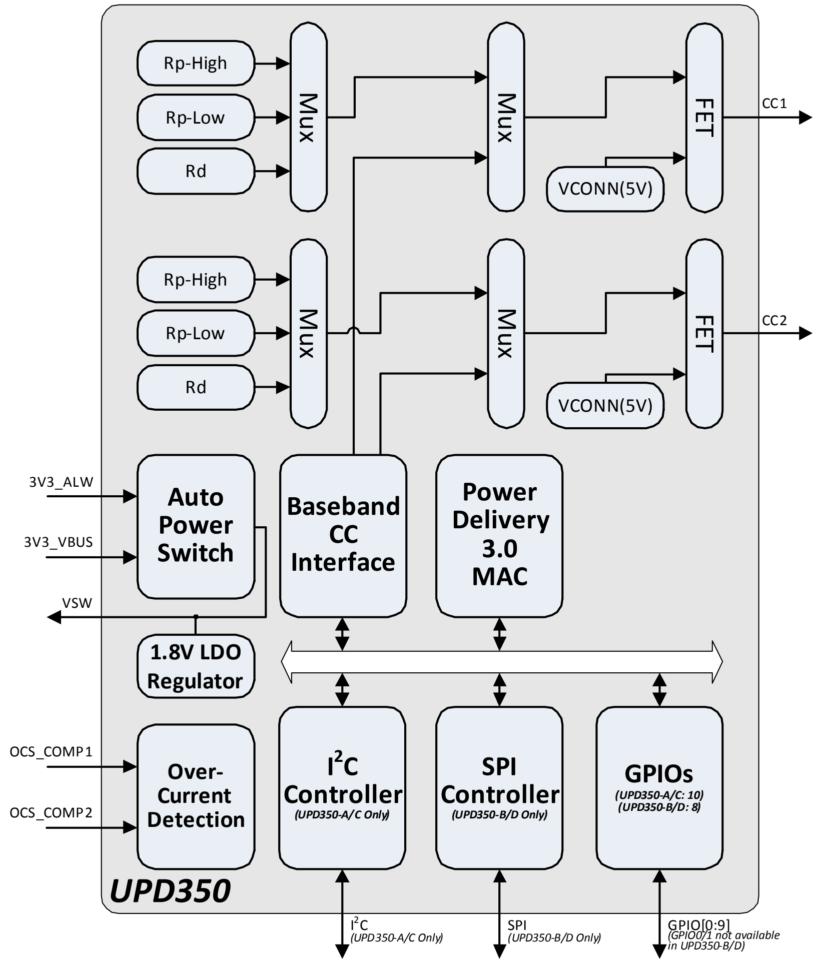

Microchip USB Hub
I^{2}C/SPI
Microchip UPD350
CC1
CC2
USB Crossbar / Alternate Mode Switches
USB Type-C^{TM} Connector

FIGURE 2-2: INTERNAL BLOCK DIAGRAM



2.2 UPD350 Family Differences Summary

The UPD350 is available in four versions:

- UPD350-A
- UPD350-B
- UPD350-C
- UPD350-D

A summary of the differences between these versions is provided in Table 2-1. Device specific features that do no pertain to the entire UPD350 family are called out independently throughout this document. For ordering information, refer to the Product Identification System on page 62.

TABLE 2-1: UPD350 FAMILY DIFFERENCES

Device	+1.8V-3.3V I^2C Interface	SPI Interface	Standalone UFP Mode	Dead Battery Support
UPD350-A	X		X	X
UPD350-B		X		X
UPD350-C	X		X	
UPD350-D		X		

3.0 PIN DESCRIPTIONS AND CONFIGURATION

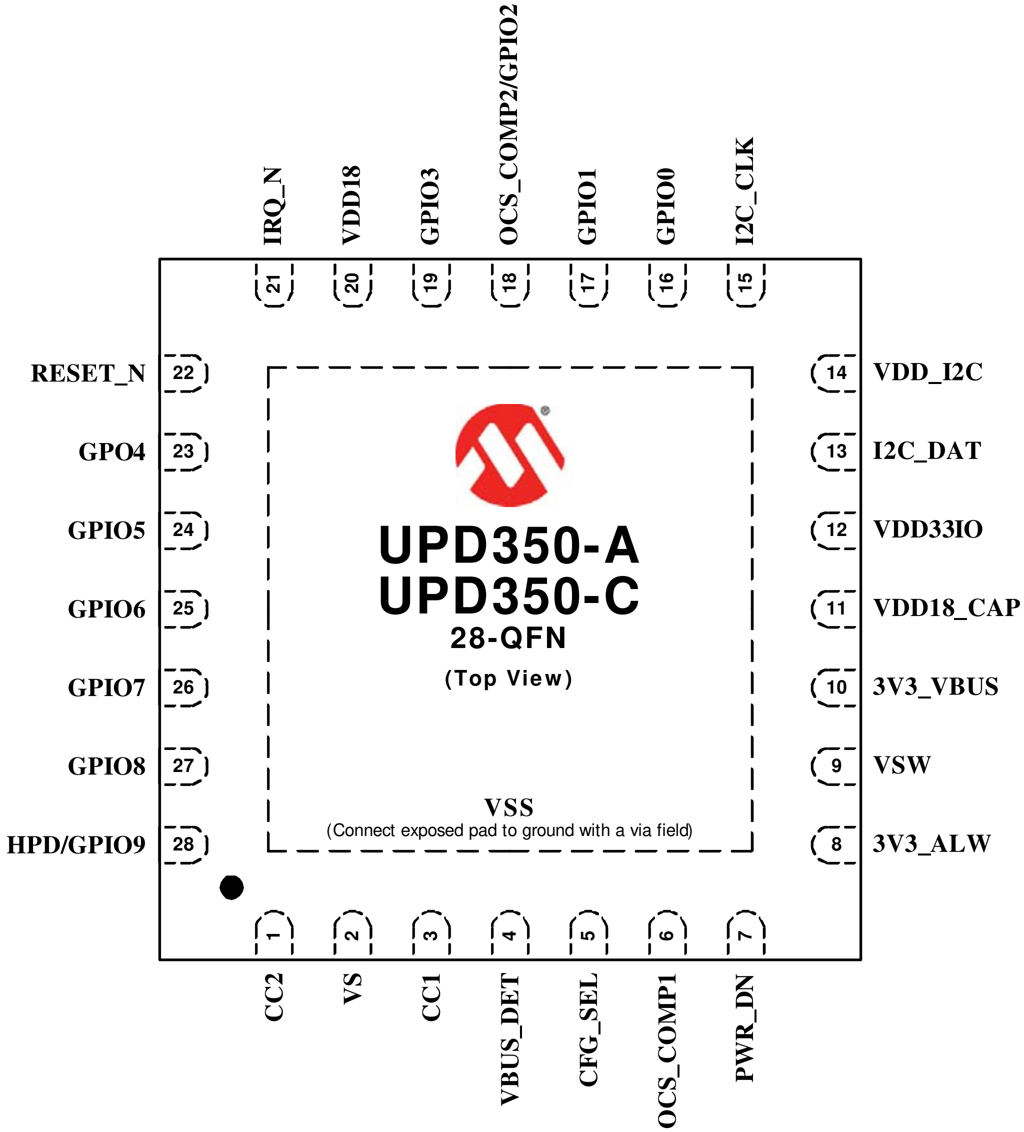
3.1 Pin Assignments

The pin assignments for the UPD350-A / UPD350-C are detailed in Section 3.1.1, "UPD350-A / UPD350-C Pin Assignments," on page 9. The pin assignments for the UPD350-B / UPD350-D are detailed in Section 3.1.2, "UPD350-B / UPD350-D Pin Assignments," on page 11. For information on the differences between the UPD350 family of devices, refer to Section 2.2, "UPD350 Family Differences Summary," on page 8.

3.1.1 UPD350-A / UPD350-C PIN ASSIGNMENTS

The pin assignments of the UPD350-A and UPD350-C devices are identical. The device pin diagram for the UPD350-A / UPD350-C can be seen in Figure 3-1. Table 3-1 provides a UPD350-A / UPD350-C pin assignment table. Pin descriptions are provided in Section 3.2, "Pin Descriptions".

FIGURE 3-1: UPD350-A / UPD350-C PIN ASSIGNMENTS (TOP VIEW)



Note: Exposed pad (VSS) on bottom of package must be connected to ground with a via field.

TABLE 3-1: UPD350-A / UPD350-C PIN ASSIGNMENTS

Pin	Pin Name	Pin	Pin Name
1	CC2	15	I2C_CLK
2	VS	16	GPIO0(Note 3-1)
3	CC1	17	GPIO1(Note 3-1)
4	VBUS_DET	18	OCS_COMP2/GPIO2(Note 3-1)
5	CFG_SEL	19	GPIO3(Note 3-1)
6	OCS_COMP1	20	VDD18
7	PWR_DN	21	IRQ_N
8	3V3_ALW	22	RESET_N
9	VSW	23	GPO4
10	3V3_VBUS	24	GPIO5(Note 3-1)
11	VDD18_CAP	25	GPIO6(Note 3-1)
12	VDD33IO	26	GPIO7(Note 3-1)
13	I2C_DAT	27	GPIO8(Note 3-1)
14	VDD_I2C	28	HPD/GPIO9(Note 3-1)

Note 3-1 This pin provides alternate functions when in Standalone UFP Mode. Refer to Section 3.1.1.1, "UPD350-A / UPD350-C GPIO Functions in Standalone UFP Modes" for additional information.

3.1.1.1 UPD350-A / UPD350-C GPIO Functions in Standalone UFP Modes

When the UPD350-A / UPD350-C is configured in Standalone UFP mode, the following GPIO pins are assigned specific alternate functions, as detailed in Table 3-2.

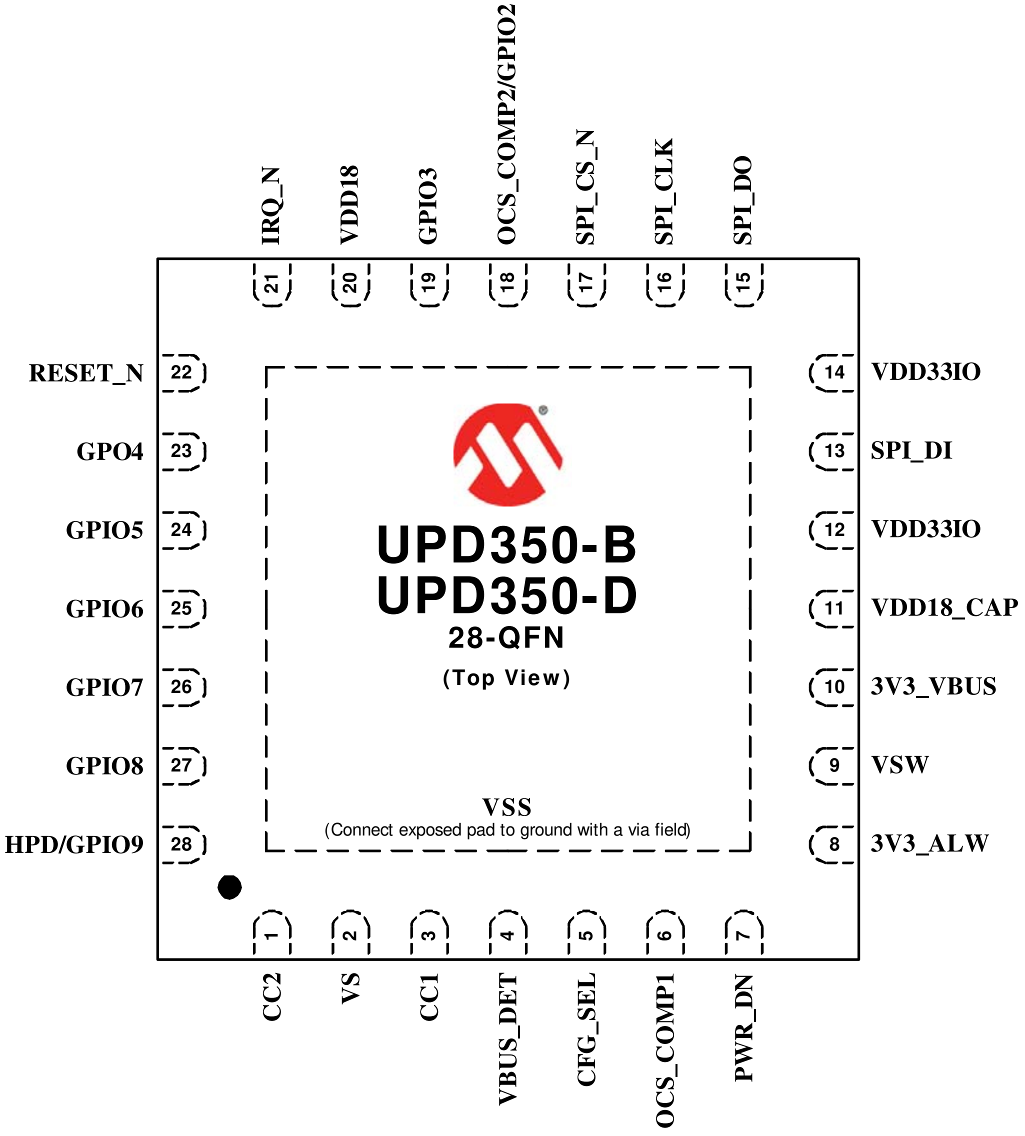
TABLE 3-2: UPD350-A / UPD350-C ALTERNATE GPIO FUNCTIONS IN STANDALONE UFP MODE

Pin	I^{2}C Companion Mode	Standalone UFP Mode
16	GPIO0	GPIO0
17	GPIO1	GPIO1
18	GPIO2	ORIENTATION
19	GPIO3	ATTACH
23	GPO4	GPO4
24	GPIO5	GPIO5
25	GPIO6	SINK_5V_LEGACY_N
26	GPIO7	SINK_5V_1A5_N
27	GPIO8	SINK_5V_3A0_N
28	GPIO9	GPIO9

3.1.2 UPD350-B / UPD350-D PIN ASSIGNMENTS

The pin assignments of the UPD350-B and UPD350-D devices are identical. The device pin diagram for the UPD350-B / UPD350-D can be seen in Figure 3-2. Table 3-3 provides a UPD350-B / UPD350-D pin assignment table. Pin descriptions are provided in Section 3.2, "Pin Descriptions".

FIGURE 3-2: UPD350-B / UPD350-D PIN ASSIGNMENTS (TOP VIEW)



Note: Exposed pad (VSS) on bottom of package must be connected to ground with a via field.

TABLE 3-3: UPD350-B / UPD350-D PIN ASSIGNMENTS

Pin	Pin Name	Pin	Pin Name
1	CC2	15	SPI_DO
2	VS	16	SPI_CLK
3	CC1	17	SPI_CS_N
4	VBUS_DET	18	OCS_COMP2/GPIO2
5	CFG_SEL	19	GPIO3
6	OCS_COMP1	20	VDD18
7	PWR_DN	21	IRQ_N
8	3V3_ALW	22	RESET_N
9	VSW	23	GPO4
10	3V3_VBUS	24	GPIO5
11	VDD18_CAP	25	GPIO6
12	VDD33IO	26	GPIO7
13	SPI_DI	27	GPIO8
14	VDD33IO	28	HPD/GPIO9

3.2 Pin Descriptions

This sections details the functions of the various device signals.

TABLE 3-4: PIN DESCRIPTIONS

Name	Symbol	Buffer Type	Description
USB Type-C™			
Configuration Channel 1	**CC1**	AIO	Configuration Channel (CC) used in the discovery, configuration and management of connections across a USB Type-C cable.
Configuration Channel 2	**CC2**	AIO	Configuration Channel (CC) used in the discovery, configuration and management of connections across a USB Type-C cable.
I^{2}C Interface *(UPD350-A / UPD350-C Only)*			
I^{2}C Clock	**I2C_CLK**	I2C	+1.8/3.3V I^{2}C clock signal
I^{2}C Data	**I2C_DAT**	I2C	+1.8/3.3V I^{2}C data signal
SPI Interface *(UPD350-B / UPD350-D Only)*			
SPI Clock	**SPI_CLK**	IS	SPI clock. The maximum supported SPI clock frequency is 25 MHz.
SPI Data Out	**SPI_DO**	O8	SPI output data.
SPI Data In	**SPI_DI**	IS	SPI input data.
SPI Chip Enable	**SPI_CS_N**	IS	Active low SPI chip enable input.
Power Delivery Control			
Hot Plug Detect	**HPD**	IS/O8	DisplayPort Hot Plug Detection.
VBUS Discharge	**DISCHARGE**	O8	VBUS discharge. **Note:** This signal is not available in the UPD350-B / UPD350-D.
Type-C Attach	**ATTACH**	O8	In the Standalone UFP mode *(UPD350-A / UPD350-C only)*, this signal indicates that the USB Type-C receptacles at the near and far end of the cable both have a plug-in. 0b: Nothing attached 1b: USB Type-C port has an end-end attached **Note:** Float this signal when unused. **Note:** This signal is not available in the UPD350-B / UPD350-D.
Type-C Orientation	**ORIENTATION**	O8	In the Standalone UFP mode *(UPD350-A / UPD350-C only)*, this signal is used to indicate which CC pin is terminated by the attached device. 0b: **CC1** pin is pulled to a higher voltage than **CC2**. 1b: **CC2** pin is pulled to a higher voltage than **CC1**. **Note:** Float this signal when unused. **Note:** This signal is not available in the UPD350-B / UPD350-D.

TABLE 3-4: PIN DESCRIPTIONS (CONTINUED)

Name	Symbol	Buffer Type	Description
Sink Legacy Current	**SINK_5V_LEGACY_N**	OD8	In the Standalone UFP mode *(UPD350-A / UPD350-C only)*, this pin asserts autonomously when a source has been detected that provides legacy USB current. **Note:** Float this signal when unused. **Note:** This signal is not available in the UPD350-B / UPD350-D.
Sink 1.5A Current	**SINK_5V_1A5_N**	OD8	In the Standalone UFP mode *(UPD350-A / UPD350-C only)*, this pin asserts autonomously when a source has been detected that provides 1.5A USB current. **Note:** Float this signal when unused. **Note:** This signal is not available in the UPD350-B / UPD350-D.
Sink 3A Current	**SINK_5V_3A0_N**	OD8	In the Standalone UFP mode *(UPD350-A / UPD350-C only)*, this pin asserts autonomously when a source has been detected that provides 3.0A USB current. **Note:** Float this signal when unused. **Note:** This signal is not available in the UPD350-B / UPD350-D.
Miscellaneous			
Interrupt	**IRQ_N**	OD8	Active low interrupt signal. **Note:** Float this signal when unused.
VBUS Detection	**VBUS_DET**	AIO	Scaled down version of VBUS. Tie this signal to VBUS via a resistor divider.
Configuration Select	**CFG_SEL**	AIO	This multi-level configuration signal is sampled after a system reset to select the device’s default mode of operation based on the connected 1% precision resistor value. **Note:** This pin is used to determine the default I^{2}C slave address and operating mode in the UPD350-A / UPD350-C. For the UPD350-B / UPD350-D, this pin can be used for customer specific purposes to provide a discrete value (0-15) based upon the attached resistor value.

TABLE 3-4: PIN DESCRIPTIONS (CONTINUED)

Name	Symbol	Buffer Type	Description
General Purpose I/O 0-9	**GPIO0, GPIO1, GPIO2, GPIO3, GPO4, GPIO5, GPIO6, GPIO7, GPIO8, GPIO9**	IS/O8/ OD8 (PU)	The general purpose I/O signals are fully programmable as either a push-pull output, an open-drain output, or a Schmitt-triggered input (except **GPO4**). A programmable pull-up may optionally be enabled. **Note:** The functionality of these GPIOs is defined and controlled by USB Power Delivery firmware executed external to the UPD350 (in the Microchip USB hub or embedded controller). **Note:** **The GPO4 general purpose signal can only function as an output and must be pulled up externally.** **Note:** Tie these signals to ground when unused. **Note:** External pull-ups and pull-downs shall be placed on GPIO pins to ensure that when in the reset state the inputs to external devices are driven to a valid state. **Note:** **GPIO0** and **GPIO1** are not available in the UPD350-B / UPD350-D. **Note:** In Standalone UFP mode *(UPD350-A / UPD350-C only)*, select GPIOs have alternate dedicated functions, as defined in Section 3.1.1.1, "UPD350-A / UPD350-C GPIO Functions in Standalone UFP Modes," on page 10.
System Reset	**RESET_N**	IS	Active low system reset. **Note:** If this signal is unused, it must be pulled up to **VDD33IO**.
Power Down	**PWR_DN**	AI	When asserted, this signal places the device into the power-down state. **Note:** Tie this signal to ground when unused.
Over-Current Sense Comparator 1	**OCS_COMP1**	AI	This pin is used by the integrated OCS comparator to detect for error conditions. **Note:** Tie this signal to ground when unused.
Over-Current Sense Comparator 2	**OCS_COMP2**	AI	This pin is used by the integrated OCS comparator to detect for error conditions. **Note:** Tie this signal to ground when unused.
Power/Ground			
+3.3V Voltage Switch Supply	**VSW**	P	+3.3V power supply output from the integrated power switch. **Note:** This pin also provides capacitance for the integrated power switch and must be connected to a 1 uF (<100 Mohm ESR) capacitor to ground.
+3.3V VBUS Supply	**3V3_VBUS**	P	+3.3V main power supply input derived from VBUS to the integrated power switch. **Note:** The 2.2 uF capacitor is only required for the UPD350-A and UPD350-B.

TABLE 3-4: PIN DESCRIPTIONS (CONTINUED)

Name	Symbol	Buffer Type	Description
+3.3V Always Supply	**3V3_ALW**	P	+3.3V main power supply input to the integrated power switch. **Note:** This pin must be connect to a 2.2 uF capacitor to ground.
+3.3V I/O Power Supply Input	**VDD33IO**	P	+3.3V I/O power supply input.
+3.3/1.8V I^{2}C Power Supply Input	**VDD_I2C**	P	+3.3/1.8V I^{2}C power supply input. Tie this pin to **VDD33IO** for +3.3V I^{2}C interfaces. Tie this pin to **VDD18** for +1.8V I^{2}C interfaces. **Note:** This pin is not available in the UPD350-B / UPD350-D.
+1.8V Core Voltage Power Supply Input	**VDD18**	P	+1.8V core voltage power supply input.
+1.8V Digital Core Power Supply Capacitor	**VDD18_CAP**	P	+1.8V digital core power supply capacitor. This signal must be connected to a 1uF capacitor to ground for proper operation.
+5V VS Power Supply Input	**VS**	P	+5V VCONN FET power source.
Ground	**VSS**	P	Ground pins.

4.0 I^{2}C SLAVE CONTROLLER *(UPD350-A/UPD350-C ONLY)*

This chapter details the integrated I^{2}C slave controller (**I2C_DAT** and **I2C_CLK**) available in the UPD350-A and UPD350-C. The I^{2}C slave controller can be used for Host CPU serial management and data transfer, and allows host access to all device Configuration and Status Registers.

4.1 I^{2}C Overview

I^{2}C is a bi-directional 2-wire data protocol. A device that is currently sending data is defined as the “transmitter” and a device that is currently receiving data is defined as the “receiver”. The bus is controlled by a master which generates the SCL clock, controls bus access, and generates the start and stop conditions. The master and slave will operate as transmitter or receiver, bit-by-bit, as determined by the master. Since the device I^{2}C controller is a slave only, the terms “host” and “master” are synonymous, both referring to the external side of the interface.

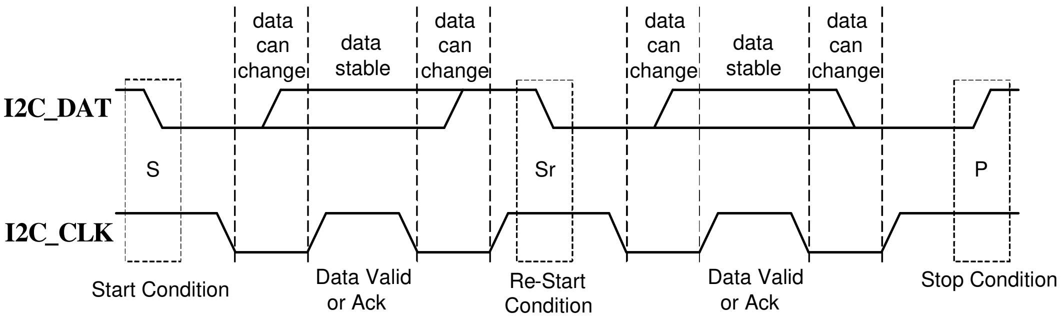
Both the clock (SCL) and data (SDA) signals have analog input filters that reject pulses that are less than 50 ns. The data pin is driven low when either interface sends a low, emulating the wired-AND function of the I^{2}C bus. Since the slave interface never drives the clock pin, the wired-AND is not necessary.

The following bus states exist:

- **Idle:** Both **I2C_DAT** and **I2C_CLK** are high when the bus is idle.
- **Start & Stop Conditions:** A start condition (S) is defined as a high to low transition on the SDA line while SCL is high. A stop condition (P) is defined as a low to high transition on the SDA line while SCL is high. The bus is considered to be busy following a start condition and is considered free 4.7 µs / 1.3 µs / 0.5µs (for 100 kHz / 400 kHz / 1MHz operation, respectively) following a stop condition. The bus stays busy following a repeated start condition (Sr) in the absence of a stop condition. Stop/start sequences and repeated starts are otherwise functionally equivalent.
- **Data Valid:** Data is valid, following the start condition, when SDA is stable while SCL is high. Data can only be changed while the clock is low. There is one valid bit per clock pulse. Every byte must be 8 bits long and is transmitted MSB first.
- **Acknowledge:** Each byte of data is followed by an acknowledge bit. The master generates a ninth clock pulse for this bit, and the transmitter releases SDA (high). To provide a positive “acknowledge” (ACK), the receiver drives SDA low so that it remains valid during the high period of the clock, taking into account the setup and hold times. To provide a negative “no-acknowledge” (NACK or ACK), the receiver will allow the line to remain high during this bit time. The receiver may be the master or the slave depending on the direction of the data. Typically the receiver acknowledges each byte. If the master is the receiver, it does not generate an acknowledge on the last byte of a transfer. This informs the slave to not drive the next byte of data, freeing SDA so that the master may generate a stop or repeated start condition.

Figure 4-1 displays the various bus states of a typical I^{2}C cycle.

FIGURE 4-1: I^{2}C CYCLES



4.2 I²C Slave Operation

The I²C slave serial interface consists of a data wire (**I2C_DAT**) and a serial clock (**I2C_CLK**). The serial clock is driven by the master, while the data wire is bi-directional. Both signals are open-drain and require external pull-up resistors.

The I²C slave controller implements the low level I²C slave serial interface (start and stop condition detection, data bit transmission/reception and acknowledge generation/reception), handles the slave command protocol and performs system register reads and writes. It tolerates and also provides clock stretching, in particular for supporting a transparent Wake on Host Access (see Section 6.3, "Asynchronous I2C Wakeup (UPD350-A/UPD350-C Only)," on page 28).

The I²C slave controller conforms to the NXP *I²C-Bus Specification* (UM10204, April 4, 2014), and supports traffic as defined therein for the following modes:

- Standard-mode (Sm, 100 kbit/s)
- Fast-mode (Fm, 400 kbit/s)
- Fast-mode Plus (Fm+, 1 Mbit/s)

Refer to Section 14.6.2, "I2C Slave Interface (UPD350-A/UPD350-C only)," on page 55 for timing information.

4.2.1 I²C SLAVE COMMAND FORMAT

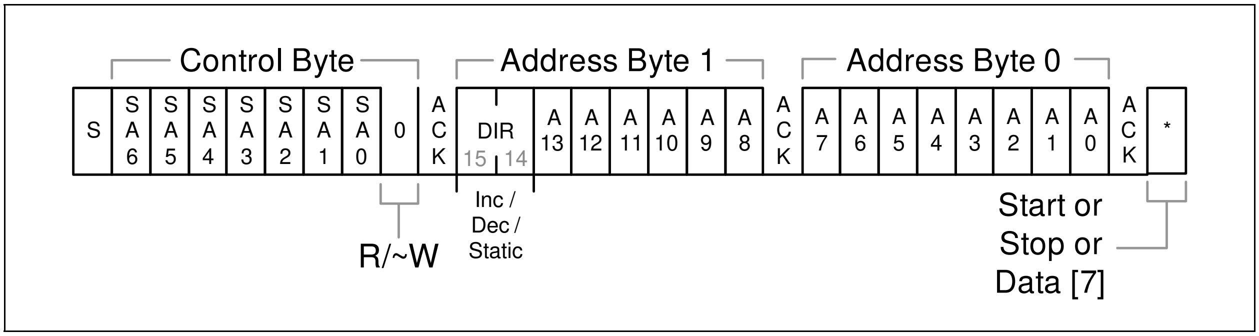
The I²C slave serial interface supports single register and multiple register Read and Write commands. A Read or Write command is started by the master first sending a Start condition, followed by a Control byte. The Control byte consists of a 7-bit slave address and a 1-bit Read/Write indication (R/~W). The default slave address used by the device is selected via the CFG_SEL configuration strap. Assuming the slave address in the Control byte matches this address, the Control byte is acknowledged by the device. Otherwise, the entire sequence is ignored until the next Start condition. The I²C slave controller also supports the General Call Address. The I²C command formats can be seen in Figure 4-2, Figure 4-4, and Figure 4-5.

If the read/write indication (R/~W) in the Control byte is a 0 (Write), the next two bytes sent by the master are a register address, and these two bytes are mandatory. The upper (first) two bits of the address field are a Direction control (DIR), which indicates whether multi-byte accesses will increment, decrement, or fix (as static) the issued address (Section 4.2.2). After the address bytes are acknowledged by the device, the master may send data bytes, which will be written to successive registers starting at this address. It may instead send another Start condition (to start the reading of data) or a Stop condition (only setting the address). The latter two will terminate the current Write before writing any data, but will have the effect of setting the internal register address which will be used for subsequent Reads.

If the read/write indication (R/~W) in the Control byte is a 1 (Read), the device will start sending data following the Control byte acknowledge bit. Read commands cannot designate an address by themselves, but may optionally be prefixed with a Write command to set it (see Figure 4-4, prefixes in gray). If however the Read immediately follows a Multiple Register Write or Read, the address may have been incremented or decremented internally according to its DIR field, so this Read will start its access at the next successive byte address. Also, regardless of the previous access, a multiple-byte Read will continue the Increment/Decrement internally, as determined by the previously-issued DIR field (Section 4.2.2).

The length of the register address field is always two full bytes. Some high-order bits are don't-care. Don't-care register address bits should be sent as '0' always, for upward compatibility.

FIGURE 4-2: I²C SLAVE ADDRESSING



> **Note:** Within bytes (address and data), the bits are transferred most-significant bit first. Addresses are transferred Most-Significant Byte first. All registers are accessed in units of bytes, and register data is transferred in increasing byte address order. Refer to the device register layout to determine the effect of this on the significance order of any multi-byte value.

4.2.2 MULTIPLE-BYTE REGISTER ADDRESS SEQUENCING

The DIR subfield in Address field bits [15:14] determines how multiple-byte sequences will be interpreted. This field is held internally whenever issued with an address, but is not applied in I^{2}C except in multiple-byte transfers, Read or Write. The DIR field definitions are as follows:

- **DIR = 00b:** Selects auto-incrementing of the internally-held register address for subsequent byte accesses in a multiple-byte packet.
- **DIR = 10b:** Selects auto-decrementing of the internally-held register address for subsequent byte accesses in a multiple-byte packet.
- **DIR = 11b:** Select a fixed address. No modification of the internal register address will occur, meaning that all subsequent accesses, single- or multiple-byte, are made to the same register.
- **DIR = 01b:** Reserved for future use.

Note that the DIR field is altered only by issuing an address. It remains, affecting any subsequent multiple-byte Read packets, until altered.

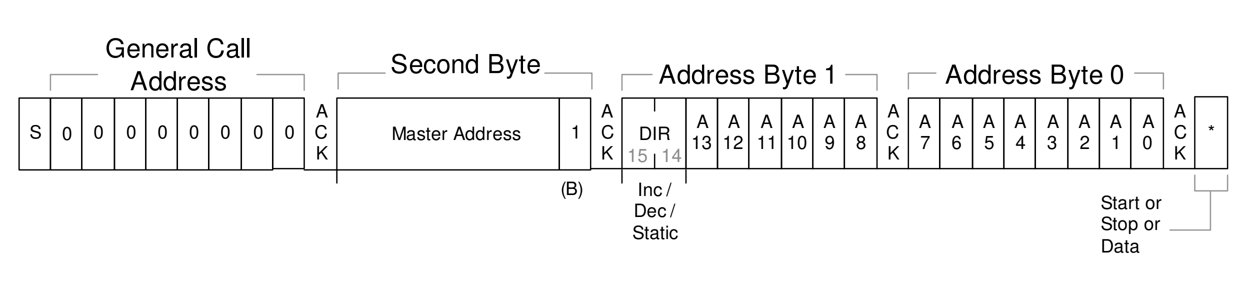
4.2.3 GENERAL CALL ADDRESS

The device supports the I^{2}C General Call Address. The intent of this feature is to enable global I^{2}C writes to topologies that have multiple UPD350 slaves. This minimizes the I^{2}C transactions for device reset, as well as for various common configuration registers. This mode of operation is intended for topologies that consist solely of UPD350 slaves. This mode of operation may not be compatible with non-UPD350 slaves coexisting on the I^{2}C bus.

Only the case where the least significant bit, “B”, of the General Call address is set to one is supported. The device will ignore the case when the least significant bit, “B”, of the General Call address is set to zero. For the latter case, the device will ACK the first byte, General call address. The device will ignore and silently discard all subsequent bytes and not acknowledge them. The second byte of the General Call address is also ignored and not acknowledged by the device.

Figure 4-3 illustrates the supported General Call Address format.

FIGURE 4-3: I^{2}C GENERAL CALL ADDRESS



4.2.4 DEVICE INITIALIZATION

Until the device has initialized itself to the point where the various configuration inputs are valid, the I^{2}C slave interface will not respond to or be affected by any external pin activity. The device should not be accessed by the master in this state. If, however, it is necessary to do so, this state will appear externally as a NACK (high) in the ACK bit time of the Control Byte, and of any further bytes transmitted by the master. The device will continue to act in this manner until the first Start condition is received after it is initialized internally. A Read transaction should not be attempted until an Address Write has been completed successfully (Figure 4-2), since the value(s) read may be unpredictable otherwise. Alternatively, an **IRQ_N** pin assertion can be used to indicate the device is ready.

4.2.5 ACCESS DURING AND FOLLOWING POWER MANAGEMENT

During low-power modes, a Start condition will trigger the device to wake, and the device will also stretch the I^{2}C clock low until its internal clocks are running and locked. It will then release the I^{2}C clock, and process the incoming packet.

It performs these steps before receiving the Slave Address bits, meaning that if there are multiple devices of this type asleep on the same I^{2}C bus segment then they will all stretch the clock, and they will all wake, regardless of whether they were actually addressed. In the event that the slave address of the I^{2}C transaction does not match the value specified in the I2C Slave Address Register (I2C_ADDR) (UPD350-A/UPD350-C Only), the device will power-down automatically.

4.2.6 I^{2}C SLAVE READ SEQUENCE

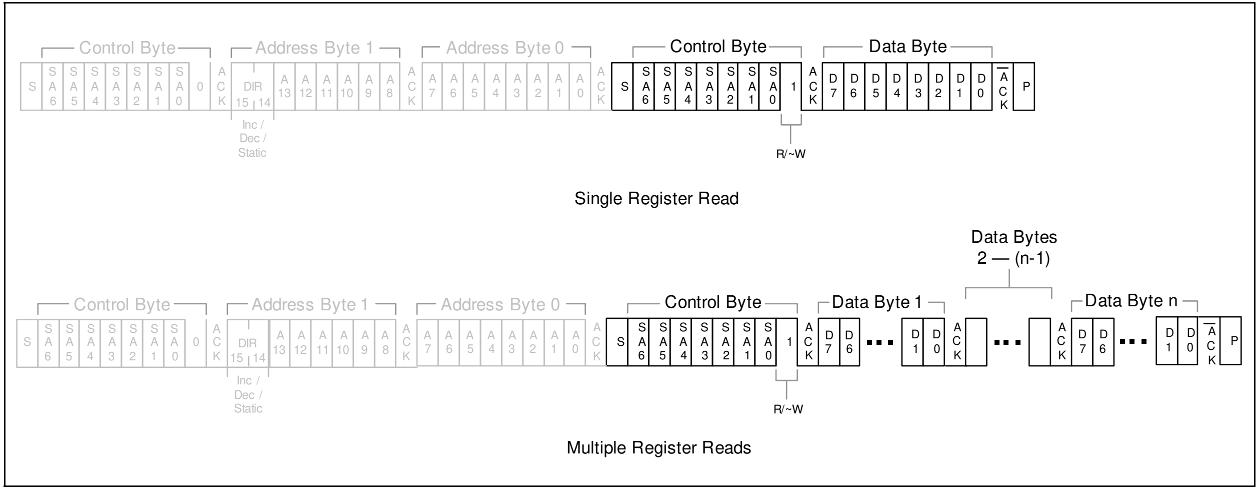
Following the device addressing, as detailed in Section 4.2.1, a register is read from the device when the master sends a Start condition and Control byte with the R/~W bit set to '1'. Assuming the slave address in the Control byte matches the device address, the Control byte is acknowledged by the device. Otherwise, the entire sequence is ignored until the next Start condition. Following the acknowledge, the device sends 1 or more bytes of data, from successive register addresses according to the last-issued DIR address subfield (Section 4.2.2), until the master sends a no-acknowledge followed by the Stop condition. The no-acknowledge informs the device not to send any further bytes.

The internal register address is unchanged if only a single register byte is read, otherwise (a Multiple Register Read) the internal register address may be incremented or decremented (Section 4.2.2) after each byte including the final one. If the internal address reaches its maximum, it rolls over to 0.

If the master sends an unexpected start or stop condition, the device will stop sending immediately and will respond to the next sequence as needed.

Figure 4-4 illustrates a typical single and multiple register read. An optional Write of an address is allowed to occur first, shown in gray. Note that this example shows an abbreviated case, where the Write does not have a Stop condition before the Read transfer's Starts. in this case, the Stop is still allowed, but not required.

FIGURE 4-4: I^{2}C SLAVE READS



4.2.7 I^{2}C SLAVE WRITE SEQUENCE

Following the device addressing, as detailed in Section 4.2.1, a register value is written to the device when the master continues to send data bytes. Each byte is acknowledged by the device. Following any data byte, after the acknowledge, the master may either send another start condition or halt the sequence with a stop condition. The internal register address is unchanged following a single-byte write.

Multiple writes are performed when the master sends additional data bytes following the first. The internal address is automatically incremented and the next register is written. Once the internal address reaches its maximum value, it rolls over to 0. The multiple write is concluded when the master sends another start or stop condition. In performing a multiple write, the internal register address may be incremented or decremented (Section 4.2.2) for each write including the final.

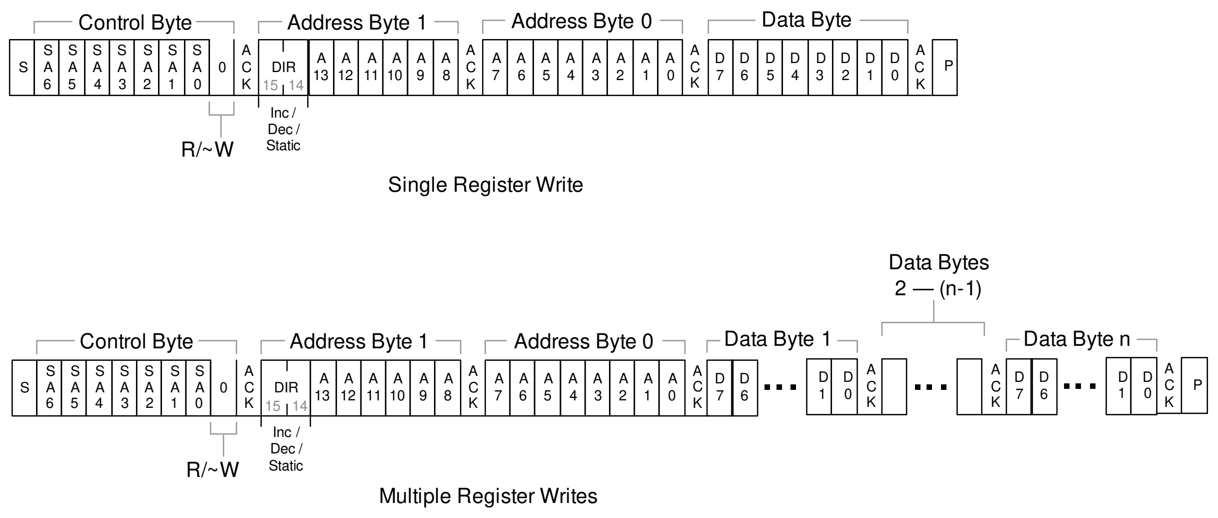
This is not relevant for subsequent writes after a new Start condition, since a new register address (with its DIR subfield) must then be included. However, this would affect the address used by any subsequent read without first resetting the register address.

For both single and multiple writes, if the master sends an unexpected start or stop condition, the device will stop immediately and will respond to the next sequence as needed.

The data write to a multi-byte register may be delayed until after all bits are input. In the event that the full register is not written (master sends a start or a stop condition occurs unexpectedly), the write may be considered invalid and the register not affected. Multiple registers may be written in a multiple write cycle, each one being written in sequence. I^{2}C writes must not be performed to unused register addresses.

Figure 4-5 illustrates a typical single and multiple register write.

FIGURE 4-5: I^{2}C SLAVE WRITES



4.2.8 SPECIAL CSR HANDLING

4.2.8.1 Live Bits

Register values are latched (registered) at the beginning of each register read to prevent the host from reading a changing value. The latching occurs individually per register in a multiple register read sequence.

4.2.8.2 Change-on-Read Registers and FIFOs

Any single-byte register that triggers a side-effect from a read operation (for example, containing “clear on read” bits, or advancing a FIFO structure) triggers only after the host has begun accessing the value. The value seen by the master will always be the original value and never the updated result of the side-effect.

For a multiple-byte register that is considered a single unit, the change may be delayed until all bytes of the register have been read. In the event that the host sends a no-acknowledge on one of the first bytes of a multi-byte register, or a start or stop condition occurs unexpectedly before the acknowledge of the full register, the read may be considered invalid and the side-effect not triggered.

4.2.8.3 Live Bits that are also Change-on-Read

As described above, the current value from a register with live bits (as is the case of any register) is captured and latched as output data, and Change on Read bits are then changed in the original register. To prevent loss of a hardware event that occurs following the data capture but before the Change on Read, these hardware events are held pending until after the read action and after any change due to the read. This sequence also ensures an edge in the bit due to the hardware event.

5.0 SPI SLAVE CONTROLLER *(UPD350-B/UPD350-D ONLY)*

This chapter details the integrated SPI slave controller (**SPI_DI**, **SPI_DO**, **SPI_CLK**, and **SPI_CS_N**) available in the UPD350-B and UPD350-D. The SPI slave controller can be used for Host CPU serial management and data transfer, and allows host access to all device Configuration and Status Registers.

5.1 SPI Overview

The SPI Slave module provides a low pin count synchronous slave interface that facilitates communication between the device and a host system. The SPI slave allows access to the System CSRs and internal FIFOs and memories. It supports single and multiple register read and write commands with incrementing, decrementing and static addressing. Only a Single bit lane is supported in SPI mode at up to 25 MHz.

The following is an overview of the functions provided by the SPI Slave:

- **Fast Read:** 4-wire (clock, select, data in and data out) reads. Serial command, address and data. This is called "Fast" Read for historical reasons, and is the only Read command supported. There is a single Dummy byte required for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- **Write:** 4-wire (clock, select, data in and data out) writes at up to 25 MHz. Serial command, address and data. Single and multiple register writes with incrementing, decrementing or static addressing.

5.2 SPI Slave Operation

A SPI frame starts on the falling edge of **SPI_CS_N**, and ends with **SPI_CS_N** rising. At the edges of **SPI_CS_N**, the **SPI_CLK** clock may be at its reset state of either low (Mode 0) or high (Mode 3), at the option of the Master.

Input data on the **SPI_DI** pin (often called "MOSI") is sampled on the rising edge of the **SPI_CLK** input clock. Output data is launched on the **SPI_DO** pin (often called "MISO") with the falling edge of the clock. While the **SPI_CS_N** chip select input is high, the **SPI_DI** and **SPI_CLK** inputs are ignored and the **SPI_DO** output is floating.

Each frame starts with an 8-bit instruction byte, transmitted by the Master, and it is accepted on **SPI_DI** starting at the first rising edge of the input clock after **SPI_CS_N** goes active.

For both Write and (Fast) Read instructions, two address bytes follow the instruction byte. The address field expresses a byte address. Fourteen address bits specify the address. The remaining two bits [15:14] constitute the DIR subfield of the address field, which specifies whether the address is Auto-Incremented (00b) or Auto-Decremented (10b) for consecutive data bytes in the frame. A special Static address coding (11b) keeps the address static throughout the frame of data, causing a single byte address to be accessed repeatedly if multiple bytes are transferred in the frame. DIR subfield encoding 01b is reserved and should be decoded in implementation to be the same as 00b, for the sake of minimizing the effect of a software error that increments beyond the address space.

For the Fast Read instruction, one dummy byte follows the address bytes. The dummy byte occupies 8 bits, one per clock.

The device will normally not drive **SPI_DO** during the Instruction, Address or Dummy byte cycles, but see Section 5.2.2, "Access During and Following Power Management," on page 23 for a special case.

For Fast Read instructions, one or more 8-bit data fields follow the dummy byte. For Write instructions, they immediately follow the address bytes.

Individual bytes in instruction, address and data fields are transferred with the most-significant bit (msb) first. The two-byte Address field is transferred with the most-significant byte (MSB) first. Multi-byte data values are transferred in the order specified by the DIR subfield of the Address field (bits [15:14]), and so their order can be effectively selected by using Increment mode (starting from the lowest byte address) or Decrement mode (starting from the highest byte address).

The SPI interface supports a minimum time of 50ns between successive commands (a minimum **SPI_CS_N** inactive time of 50ns).

The instructions supported by the SPI slave controller are listed in Table 5-1. Unsupported instructions are reserved and must not be used.

TABLE 5-1: SPI INSTRUCTIONS

Instruction	Description	Bus Bit Width	Inst. Code	Address Bytes	Dummy Bytes	Data bytes	Max Freq.
Read							
FASTREAD	Read, higher speed format	1	0Bh	2	1	1 to ∞	25 MHz
Write							
WRITE	Write	1	02h	2	0	1 to ∞	25 MHz

5.2.1 DEVICE INITIALIZATION

Until the device has been initialized to the point where the various configuration inputs are valid, the SPI interface will not respond to or be affected by any external pin activity.

Once device initialization completes, the SPI interface will ignore the pins until a rising edge of **SPI_CS_N** is detected.

If the device initialization completes during an active cycle (**SPI_CS_N** low), the trailing end of the frame must be seen (**SPI_CS_N** returning high) before any internal registers are affected or the state of the SPI interface changes.

The first SPI access after device initialization must always be a dummy read to the SPI Test Register (SPI_TEST) (UPD350-B/UPD350-D Only).

5.2.1.1 SPI Slave Read Polling for Initialization Complete

With an external weak pull-up resistor present on **SPI_DO**, a value of FFh will appear to have been read from any internal register while the device is uninitialized. By verifying the SPI Test Register (SPI_TEST) (UPD350-B/UPD350-D Only) has at least one “0” bit in it, it is possible to tell when the device is initialized.

5.2.2 ACCESS DURING AND FOLLOWING POWER MANAGEMENT

The Wake event on SPI traffic is local to the specific device, and does not affect the states of other devices even on the same SPI bus. Until waking is complete, the SPI interface holds the **SPI_DO** pin low for the duration of the **SPI_CS_N** low time.

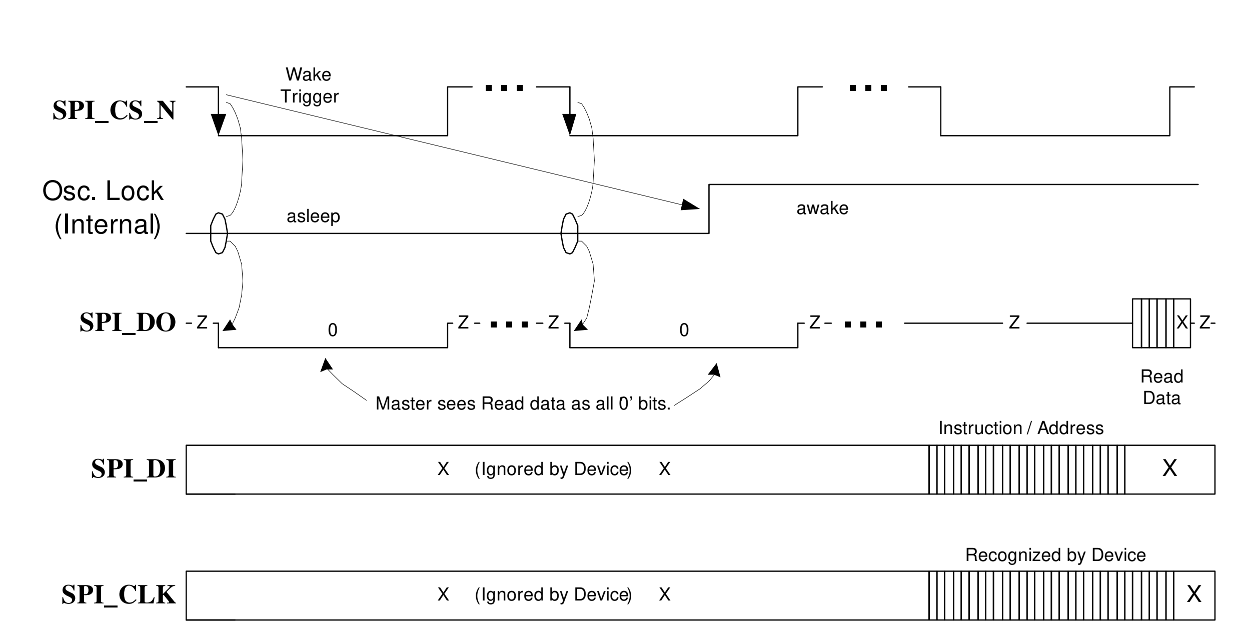
Until the device is awake, then, any Read access performed by the Master will appear to have returned all “1” bits. To determine when the device is awake and the SPI interface functional, the SPI Test Register (SPI_TEST) (UPD350-B/UPD350-D Only) should be repeatedly polled by the Master in separate frames (**SPI_CS_N** low then high). Once a correct, non-zero value is read, the interface can be considered functional. As an alternative to polling, an **IRQ_N** pin assertion can be used to indicate the device is ready.

Once the power management mode changes back to ACTIVE, the SPI interface will still ignore the **SPI_CLK** and **SPI_DI** pins, following **SPI_CS_N** low with **SPI_DO** low, until **SPI_CS_N** is seen high. At the next **SPI_CS_N** falling edge, SPI communication will continue normally.

At any time after performing SPI traffic, the device will not go back to a non-communicating power state until explicitly allowed to do so by a command from the SPI Master.

Figure 5-1 illustrates the sequence of waking from SPI traffic.

FIGURE 5-1: POWER MANAGEMENT WAKE ON SPI TRAFFIC



5.2.3 SPI READ COMMAND (FAST READ)

The Fast Read command is supported by the SPI slave. A single byte, or multiple bytes, may be read in a single frame (**SPI_CS_N** low).

Fast Read is the only form of Read access supported by the device. The instruction inputs the instruction code, the address and a dummy byte on **SPI_DI**, and outputs the data one bit per clock on **SPI_DO**.

The SPI slave interface is selected by first bringing **SPI_CS_N** active. The 8-bit FASTREAD instruction, 0Bh, is input into the **SPI_DI** pin, followed by the two address bytes and 1 dummy byte. The address bytes specify a Byte register address within the device, and also specify how addresses are sequenced for successive bytes in a Multiple Byte Read (below). The contents of the dummy byte are don't-care.

On the falling clock edge following the rising edge of the last dummy bit, the **SPI_DO** pin is driven starting with the most significant bit of the selected register byte. The remaining register bits are shifted out on subsequent falling clock edges.

The **SPI_CS_N** input is brought inactive to conclude the cycle. The **SPI_DO** pin is floated by the device in response.

5.2.3.1 Multiple Byte Reads

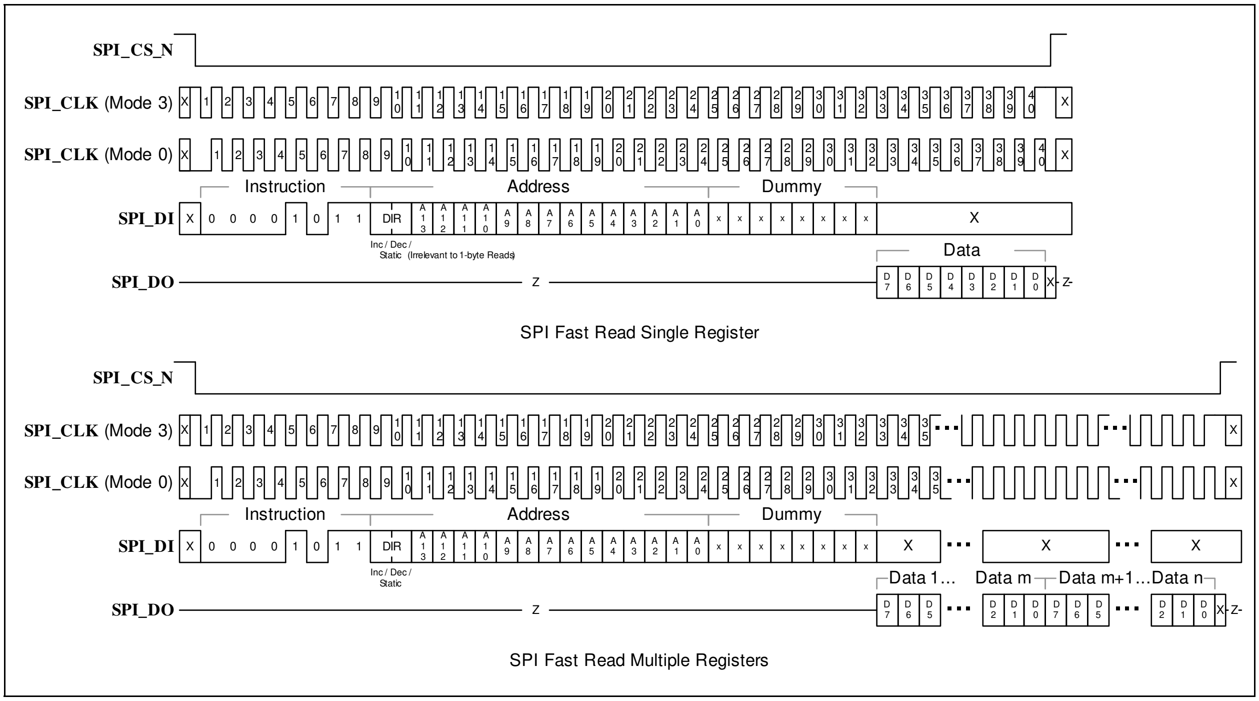
Additional byte reads beyond the first are performed by the Master by continuing the clock pulses while **SPI_CS_N** is active. The upper two bits [15:14] (DIR subfield) of the address specify Auto-Incrementing (DIR=00b) or Auto-Decrementing (DIR=10b) or Static (fixed) (DIR=11b) for successive bytes read. Maintaining a Static internal address is provided for FIFO Read/Write or low-level register polling within a single frame, if the Master supports it.

Towards the end of the current one-byte output shift the address is incremented or decremented, if appropriate, and another synchronized capture sequence is done.

5.2.3.2 Fast Read

Figure 5-2 illustrates a typical single and multiple register fast read for SPI mode.

FIGURE 5-2: SPI FAST READ


SPI_CS_N
SPI_CLK (Mode 3)
SPI_CLK (Mode 0)
Instruction
Address
Dummy
SPI_DI
X 0 0 0 0 1 0 1 1 DIR A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 x x x x x x x x X
Inc / Dec / Static (Irrelevant to 1-byte Reads)
Data
SPI_DO
Z
D7 D6 D5 D4 D3 D2 D1 D0 X Z
SPI Fast Read Single Register
SPI_CS_N
SPI_CLK (Mode 3)
SPI_CLK (Mode 0)
Instruction
Address
Dummy
SPI_DI
X 0 0 0 0 1 0 1 1 DIR A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 x x x x x x x x X ... X ... X
Inc / Dec / Static
Data 1...
Data m
Data m+1...Data n
SPI_DO
Z
D7 D6 D5 ... D2 D1 D0 D7 D6 D5 ... D2 D1 D0 X Z
SPI Fast Read Multiple Registers

5.2.4 SPI WRITE COMMANDS

The following write commands are supported by the SPI slave controller:

- Write
- Multiple Writes

5.2.4.1 Write

The Write instruction provides the instruction code and address and data bytes on the **SPI_DI** pin, one bit per clock.

The SPI transfer is started by the Master first driving **SPI_CS_N** active. The 8-bit WRITE instruction, 02h, is given on the **SPI_DI** pin, followed by the two address bytes. The address bytes specify a byte address within the device, and a Direction control subfield (DIR).

The data immediately follows the address bytes on the **SPI_DI** pin, starting with the most significant bit of the first byte. The data is input from the **SPI_DI** pin by the device, shifted in on each subsequent rising clock edge.

5.2.4.2 Multiple Writes

Multiple writes are performed by the Master by continuing the clock pulses and **SPI_DI** data while **SPI_CS_N** remains active. The upper two bits [15:14] of the address constitute the DIR subfield, and specify auto-incrementing (DIR=00b) or auto-decrementing (DIR=10b) or Static addressing. The internal Byte address is incremented, decremented, or kept fixed (Static) based on these bits. Maintaining a fixed internal address may be useful for FIFO access, register “bit-banging” or other repeated activity.

The data write to the register occurs after the full register contents are input: this depends on the defined size of the register. In the event that the full register is not written when **SPI_CS_N** is returned high, the write is considered invalid and the register is not affected.

The **SPI_CS_N** input is then brought inactive to conclude the cycle.