USB 3382-AA/AB
PCI Express Gen 2 to USB 3.0 SuperSpeed Peripheral Controller
Data Book

Version 1.3

July 2012

Website www.plxtech.com
Technical Support www.plxtech.com/support
Phone 800 759-3735
408 774-9060
FAX 408 774-2169

Revision History

Version	Date	Description of Changes
1.0	January, 2012	Production release, Silicon Revision AA.
1.2	March, 2012	Production update, Silicon Revision AA. Corrected function of pins A26 and B19 to be VAUX_IO. Previous versions of this data book incorrectly labeled these pins as VDD_IO. **Designs that power VAUX_IO and VDD_IO supplies separately are impacted by this change. Designs that connect VAUX_IO and VDD_IO supplies together are not affected.** Added note to USB_VBUS pin regarding possible leakage current. Applied miscellaneous corrections and notes throughout the data book.
1.3	July, 2012	Production release, Silicon Revision AB. Production update, Silicon Revision AA. Corrected pin locations in Figure 2-2. Added I2C pin header information to Section 2.4.6. Changed "SMBus Slave Address" to "SMBus Device Address." Cleaned up PM D-state references. Significantly updated Chapter 8, "USB Controller Functional Description." Changed register offset 31Ch to ***reserved***. Corrected the **SETUPDW0** register *Setup Byte 0* field (USB Controller, offset 98h[7:0]). Applied miscellaneous corrections and enhancements throughout the data book.

Copyright Information

Document Number: 3382-AA/AB-SIL-DB-P1-1.3

Preface

The information in this data book is subject to change without notice. This PLX data book to be updated periodically as new information is made available.

Audience

This data book provides functional details of PLX Technology's USB 3382-AA/AB PCI Express Gen 2 to USB 3.0 SuperSpeed Peripheral Controller, for hardware designers and software/firmware engineers. The information provided pertains to both Silicon Revisions (AA and AB), unless specified otherwise.

Supplemental Documentation

This data book assumes that the reader is familiar with the following documents:

- PLX Technology, Inc., www.plxtech.com

  The PLX USB 3382 Toolbox includes this data book and other supporting documentation, *such as* errata, and design and application notes.
- The Institute of Electrical and Electronics Engineers, Inc. (IEEE), www.ieee.org
  - *IEEE Standard 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture*
  - *IEEE Standard 1149.1a-1993, IEEE Standard Test Access Port and Boundary-Scan Architecture*
  - *IEEE Standard 1149.1-1994, Specifications for Vendor-Specific Extensions*
  - *IEEE Standard 1149.6-2003, IEEE Standard Test Access Port and Boundary-Scan Architecture Extensions*
- Intel Corporation, www.intel.com
  - *PHY Interface for the PCI Express Architecture, Version 2.00*
- NXP Semiconductors, ics.nxp.com
  - *The I2C-Bus Specification, Version 2.1*
- PCI Special Interest Group (PCI-SIG), www.pcisig.com
  - *PCI Local Bus Specification, Revision 3.0*
  - *PCI Bus Power Management Interface Specification, Revision 1.2*
  - *PCI Code and ID Assignment Specification, Revision 1.2*
  - *PCI to PCI Bridge Architecture Specification, Revision 1.2*
  - *PCI Express Base Specification, Revision 1.1*
  - *PCI Express Base Specification, Revision 2.0*
  - *PCI Express Base Specification, Revision 2.0 Errata*
  - *PCI Express Base Specification, Revision 2.1*
  - *PCI Express Card Electromechanical Specification, Revision 2.0*
  - *PCI Express Mini Card Electromechanical Specification, Revision 1.1*
  - *PCI Express Architecture PCI Express Jitter and BER White Paper, Revision 1.0*

- Personal Computer Memory Card International Association (PCMCIA), www.pcmcia.org
  - *ExpressCard Standard Release 1.0*
- PXI System Alliance (PXI), www.pxisa.org
  - *PXI-5 PXI Express Hardware Specification, Revision 1.0*
- SBS Implementers Forum, smbus.org
  - *System Management Bus (SMBus) Specification, Version 2.0*
- USB Implementers Forum, www.usb.org
  - *Universal Serial Bus Specification Revision 2.0*
  - *Universal Serial Bus Specification Revision 3.0*

Note: *In this data book, shortened titles are associated with the previously listed documents. The following table lists these abbreviations.*

Abbreviation	Document
PCI r3.0	*PCI Local Bus Specification, Revision 3.0*
PCI Power Mgmt. r1.2	*PCI Bus Power Management Interface Specification, Revision 1.2*
PCI-to-PCI Bridge r1.2	*PCI to PCI Bridge Architecture Specification, Revision 1.2*
PCI Express Base r1.0a	*PCI Express Base Specification, Revision 1.0a*
PCI Express Base r1.1	*PCI Express Base Specification, Revision 1.1*
PCI Express Base r2.0	*PCI Express Base Specification, Revision 2.0*
PCI Express Base r2.1	*PCI Express Base Specification, Revision 2.1*
PCI ExpressCard CEM r2.0	*PCI Express Card Electromechanical Specification, Revision 2.0*
PCI ExpressCard Mini CEM r1.1	*PCI Express Mini Card Electromechanical Specification, Revision 1.1*
IEEE Standard 1149.1-1990	*IEEE Standard Test Access Port and Boundary-Scan Architecture*
IEEE Standard 1149.6-2003	*IEEE Standard Test Access Port and Boundary-Scan Architecture Extensions*
I^2C Bus v2.1 *I2C Bus v2.1*[a]	*The I^2C-Bus Specification, Version 2.1*
SMBus v2.0	*System Management Bus (SMBus) Specification, Version 2.0*
USB r2.0	*Universal Serial Bus Specification r2.0*
USB r3.0	*Universal Serial Bus Specification r3.0*

a. Due to formatting limitations, the specification name may appear without the superscripted “2” in its title.

Terms and Abbreviations

The following tables list common terms and abbreviations used in this data book. Most terms and abbreviations defined in the *PCI Express Base r2.1* and/or *USB r3.0* are generally not included in this table.

Terms and Abbreviations	Definitions
8b/10b	Data-encoding scheme used on data transferred across a Link that is operating at either Gen 1 or Gen 2 Link speed (2.5 or 5.0 GT/s, respectively).
ACK	Acknowledge Control Packet. A Control packet used by a destination to acknowledge Data packet receipt. A signal that acknowledges signal receipt.
AEC	Auto-Enumerate Controller.
Agent	Entity that operates on the PCI Express interface.
ARI	Alternative Routing-ID Interpretation.
ARP	Address Resolution Protocol.
BAR	Base Address register.
Big Endian	Most significant byte in a scalar is located at Address 0.
BER	Bit error rate.
BIST	Built-In Self Test.
CDR	Clock/Data Recovery circuit.
Clock cycle	One period of the PCI Express interface clock.
Configurable Endpoint	Available for general Data transfers between the USB Host and PCI Express interface.
Control Transfer	Support configuration/command/status type communication to Endpoint 0 (EP 0) CSR Handler, Interrupts, Messages, and Serial EEPROM Handler.
CRC	Cyclic Redundancy Check.
CSR	Configuration Space register.
Dedicated Endpoint	Used for a pre-defined task within the USB 3382.
Device Endpoint	A uniquely addressable portion of a USB device that is the source or sink of information in a communication flow between the Host and USB 3382. The USB 3382 has six Dedicated endpoints, and up to eight general-purpose endpoints.
DLL	Data Link Layer.
DMA	Direct Memory Access.
Downstream	The direction of data flow that moves away from the Host. The USB 3382 always connects to a downstream Port.
Downstream Device	Device that is connected to a downstream Port.
Downstream Port	Port that is used to communicate with a device below it in the system hierarchy. A bridge can have one or more downstream Ports.

Terms and Abbreviations	Definitions
ECC	Error-Correcting Code.
EIES	Electrical Idle Exit Sequence.
EIOS	Electrical Idle Ordered-Set.
Electrical Idle	Transmitter is in a High-Impedance state (+ and - are both at common mode voltage).
EOP	End of Packet.
EP	Endpoint.
FC	Flow Control.
Field	Multiple register bits that are combined for a single function.
Frame	1-µs timebase established on Full-Speed Buses.
FTS	Fast Training Sequence.
Function	USB device that provides a capability to the Host.
Gbps	Gigabits per second.
Gen 1	*PCI Express Base r1.1* and below. Link transfer rate of 2.5 GT/s.
Gen 2	*PCI Express Base r2.0* and *PCI Express Base r2.1*. Link transfer rate of 5.0 GT/s.
GPIO	General-Purpose Input/Output.
GT/s	Giga-Transfers per second.
Handshake Packet	Packet that acknowledges or rejects a specific condition.
High Bandwidth Endpoint	High-Speed device endpoint that transfers more than 1,024 bytes and less than 3,073 bytes, per microframe.
Host	The Host computer system in which the USB Host Controller is installed. This includes the Host hardware platform and operating system in use.
HCSL	High-Speed Current Steering Logic.
HCSLOUT	HCSL Output clocks.
INCH	Ingress Credit Handler.
InitFC	Initialization Flow Control.
IRAM	Internal Random Access Memory (IRAM).
Isochronous Data	A stream of data whose timing is implied by its delivery rate.
Isochronous Transfer	Provide periodic continuous communication between Host and device.
JTAG	Joint Test Action Group.
Lane	Bidirectional pair of differential PCI Express I/O signals.
LC	Inductor Capacitor.
LCRC	Link Cyclic Redundancy Check.
Link	Active connection between two Ports or devices.
Little Endian	Least significant byte in a scalar is located at Address 0.
Local	Reference to PCI Express attributes (*such as* credits) that belong to the PCI Express Link logic.
LTSSM	Link Training and Status State Machine.
LVDS	Low-Voltage Differential Signaling.
LVPECL	Low-Voltage Positive Emitter-Coupled Logic.

Terms and Abbreviations	Definitions
Mbps	Megabits per second.
MBps	Megabytes per second.
MCU	Micro-Controller Unit (8051).
Message Pipe	Bidirectional pipe that transfers data using a request/data/status paradigm. The data has an imposed structure that allows Requests to be reliably identified and communicated.
Microframe	125-µs time base established on High-Speed USB Buses.
MIPs	Million instructions per second.
MPS	Maximum Payload Size.
NACK	Negative Acknowledge. Used in the SMBus-related content.
NAK	Negative Acknowledge.
N_FTS	Number (quantity) of Fast Training Sequences field in Training Sets.
NOP	No Operation.
NRDY	Not Ready.
OS	Ordered-Set.
Packet	Bundle of data organized in a group for transmission. Packets typically contain control information, data, and error detection/correction bits.
Packed ID (PID)	Field in a USB packet that indicates the type of packet, and by inference, the format of the packet and type of error detection applied to the packet.
PEC	Packet Error Code.
PEX	PCI Express.
PHY	Physical Layer.
PIPE	PHY Interface for PCI Express architecture.
Pipe	A logical abstraction representing the association between an endpoint on a USB device and software on the Host.
PLL	Phase-Locked Loop.
PM	Power Management.
PME	Power Management Event.
PN	Port Number.
Port	Interface to a group of SerDes and supporting logic that is capable of creating a Link, for communication with another Port.
Port ID	Number, assigned in hardware, that associates a SerDes with a Port.
PRBS	Pseudo-Random Bit Sequence.
QoS	Quality of Service.
RAS	Reliability, Availability, and Serviceability.
RoHS	Restrictions on the use of certain Hazardous Substances (RoHS) Directive.
Root Port	Downstream Port on a Root hub.
Rx	Receiver.

Terms and Abbreviations	Definitions
Sample	Smallest unit of data upon which an endpoint operates.
Scalar	Multi-byte data element.
SE0	Single-ended zero.
SerDes	Serializer/De-Serializer. A high-speed differential-signaling parallel-to-serial and serial-to-parallel conversion logic attached to Lane pads.
SN	SerDes Number.
SPI	Serial Peripheral Interface.
SSC	Spread-Spectrum Clock.
Start of Frame (SOF)	First transaction in each USB (micro)frame. An SOF allows endpoints to identify the start of the (micro)frame and synchronize internal endpoint clocks to the Host.
Sticky Bits	Register bits in which the current values are unchanged by a Hot Reset, Link Down event or a Secondary Bus Reset, while the USB 3382 is powered. Sticky bits are reset to default values by a Fundamental Reset. HwInit, ROS, RW1CS, and RWS CSR types. (Refer to Table 15-6, "Register Types, Grouped by User Accessibility," and Table 16-1, "Access Attributes," for CSR type definitions.)
Sticky State	Condition that causes a state machine to be stuck in a particular state, unable to make forward progress.
TC	Traffic Class.
TCB	Training Control Bits field in Training Sets.
TL	Transaction Layer.
TLC	Transaction Layer Control. The module performing PCI Express Transaction Layer functions.
TLP	Transaction Layer Packet. PCI Express packet formation and organization.
Token Packet	Type of packet that identifies what transaction is to be performed on the bus.
Transaction	Delivery of service to an endpoint. Consists of a Token packet, optional Data packet, and optional Handshake packet. Specific packets are allowed/required, based upon the transaction type.
Transfer	One or more Bus transactions to move information between a software client and its function.
TS1	Type 1 Training Sequence Ordered-Set.
TS2	Type 2 Training Sequence Ordered-Set.
Turnaround Time	Length of time a device must wait before starting to transmit a packet, after a packet is received, to prevent collisions on the USB.
Tx	Transceiver.
UDID	Unique Device Identifier.
UI	Unit Interval – 400 ps at 2.5 GT/s, 200 ps at 5.0 GT/s.
Upstream	Direction of data flow toward the Host.
Upstream Device	Device that is connected to Port 0.
Upstream Port	Port 0, used to communicate with a device above it in the system hierarchy. Electrically closest to the Host. Receives downstream data traffic.
USB	Universal Serial Bus.
UTP	User Test Pattern.

Terms and Abbreviations	Definitions
VC	Virtual Channel. The USB 3382 supports one Virtual Channel, VC0.
Vector	Address and data.
ZLP	Zero-length packet.

Data Book Notations and Conventions

Notation / Convention	Description
Blue text	Indicates that the text is hyperlinked to its description elsewhere in the data book. Left-click the blue text to learn more about the hyperlinked information. This format is often used for register names, register bit and field names, register offsets, chapter and section titles, figures, and tables.
PEX_XXXn[*x*] PEX_XXXp[*x*]	When the signal name appears in all CAPS, with the primary Port description listed first, field [*x*] indicates the number associated with the signal pins/pads assigned to a specific SerDes module/Lane. The lowercase “n” (negative) or “p” (positive) suffix indicates the differential pair of signals, which are always used together.
# = Active-Low signals	Unless specified otherwise, Active-Low signals are identified by a “#” appended to the term (*for example*, PEX_PERST#).
`Program/code samples`	`Monospace font` (*`program or code samples`*) is used to identify code samples or programming references. These code samples are case-sensitive, unless specified otherwise.
Command/Status	Register names.
Parity Error Detected	Register parameter [bit or field] or control function.
Upper Base Address[31:16]	Specific Function in 32-bit register bounded by bits [31:16].
Number multipliers	k = 1,000 (10^3) is generally used with frequency response. K = 1,024 (2^{10}) is used for Memory size references. KB = 1,024 bytes. M = meg. = 1,000,000 when referring to frequency (decimal notation) = 1,048,576 when referring to Memory sizes (binary notation)
255d	d = Suffix that identifies decimal values.
1Fh	h = Suffix that identifies hex values. Each prefix term is equivalent to a 4-bit binary value (Nibble). Legal prefix terms are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
1010b	b = suffix which identifies binary notation (*for example*, 01b, 010b, 1010b, and so forth). Not used with single-digit values of 0 nor 1.
0 through 9	Decimal numbers, or single binary numbers.
byte	Eight bits – abbreviated to “B” (*for example*, 4B = 4 bytes).
LSB	Least-Significant Byte.
lsb	Least-significant bit.
MSB	Most-Significant Byte.
msb	Most-significant bit.
DWord or DW	Double-Word (32 bits) is the primary register size in these devices.
QWord	Quad-Word (64 bits).
Reserved	Do not modify ***reserved*** register bits and fields. Unless specified otherwise, these bits read as 0 and must be written as 0.
word	16 bits.

Contents

Registers

PCI Configuration Registers

USB Configuration Registers

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 1 Introduction

1.1 Overview

This data book describes PLX Technology's USB 3382 PCI Express Gen 2 to USB 3.0 SuperSpeed Peripheral Controller. The USB 3382 features two PCI Express x1 Gen 2 Ports and one *USB r3.0* SuperSpeed client Port.

The USB 3382 provides a matching bandwidth at 5 Gbps between the PCI Express Gen 2 interface and *USB r3.0* SuperSpeed Bus. The USB 3382 can easily add a SuperSpeed USB client Port to an existing PCI Express system, as well as convert existing PCI Express functions (endpoints) to a SuperSpeed USB product.

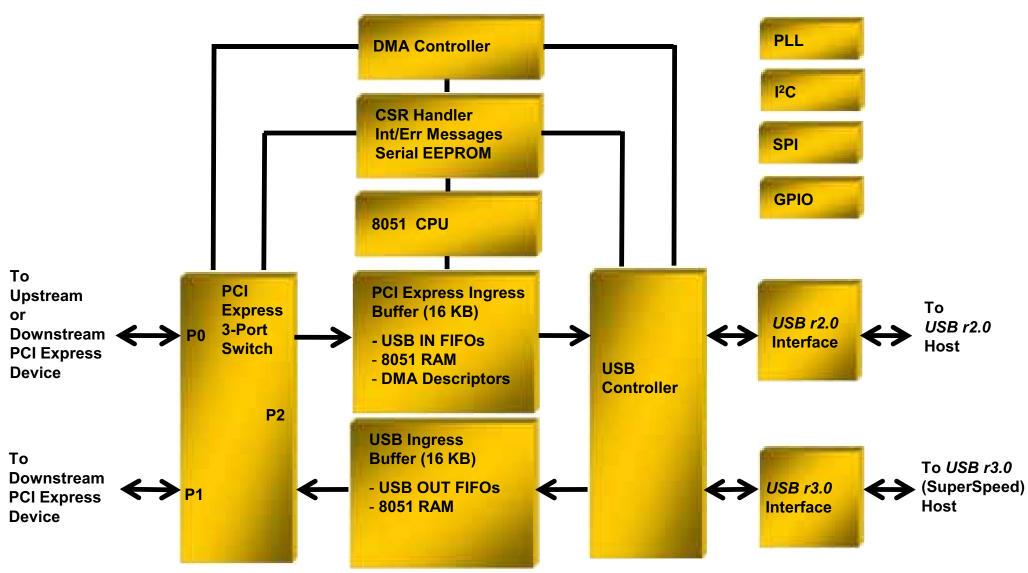
The USB 3382 features an internal high-performance switch that can configure its two PCI Express Ports into one x1 upstream Port and one x1 downstream Port, or two x1 downstream Ports, or one x2 upstream/downstream Port. The flexibility allows different system configurations to achieve the maximum performance of the product.

As the successor of the gold standard PLX NETCHIP™ NET 2282 PCI to High-Speed USB 2.0 Controller, the USB 3382 can be used with existing NET 2282 software, with no or minimal change. Driver stacks are already available in common Operating Systems/Environments, *such as* Microsoft Windows (XP, Vista, 7, and CE), Linux, and VxWorks. PLX's USB Duet® software provides a PC interconnect at 400 MBps, with just a simple USB cable.

Target applications for the USB 3382 as a PCI Express Adapter (endpoint) include PCs, servers, PCI Express Adapter (endpoint) docking stations, printers, and PCI Express embedded systems. The main applications for the USB 3382 as a PCI Express Root Complex include WLAN dongles, graphics/video dongles, and HDTV tuners/codecs.

A functional block diagram of the USB 3382 is shown in Figure 1-1.

Figure 1-1. USB 3382 Block Diagram



1.2 Features

The USB 3382 supports the following features:

- Two pairs of buffered, 100-MHz HCSL output clocks, one pair for each of its downstream PCI Express Ports
- Bridges between a PCI Express interface and USB Host
- Two PCI Express x1 Gen 2 Ports
  - Operates as a PCI Express Root Complex or Endpoint
  - PCI Express switch functionality in Endpoint mode allows one upstream PCI Express Port and one downstream PCI Express Port
- One *USB r3.0* SuperSpeed peripheral Port
- Four-channel DMA Scatter/Gather Controller
- Integrated 8051 Micro-Controller Unit (MCU) (referred to herein, as *8051*) with 32-KB SRAM Program/Data memory
- USB SuperSpeed (5 Gbps), High-Speed (480 Mbps), and Full-Speed (12 Mbps) modes
- USB Auto-Enumerate modes
- Automatic Retry of failed packets
- Maximum Payload Size – 2,048 bytes
- Quality of Service (QoS) support
  - All Ports support one, full-featured Virtual Channel (VC0)
  - All Ports support eight Traffic Class (TC[7:0]) mapping, independently of the other Ports
- 4 General-Purpose Input/Output (GPIO) pins with Pulse Width Modulation (PWM)
- Other PCI Express Capabilities
  - Transaction Layer Packet (TLP) Digest support for Poison bit
  - Lane reversal (Port 0 only, when Port 0 is configured with a x2 Link width)
  - Polarity reversal
  - Conventional PCI-compatible Link Power Management states
    - L0, L0s, L1, L2, and L2/L3 Ready
    - L3 (with Vaux supported)
  - Conventional PCI-compatible Device Power Management states
    - D0, D1, D2, and D3hot
    - D3cold (with Vaux supported)
  - Active State Power Management (ASPM)
  - Dynamic Link speed (2.5 or 5.0 GT/s) negotiation
  - Dynamic Link width negotiation
- Out-of-Band Initialization options
  - Serial EEPROM
  - I^2C and SMBus (7-bit Slave address with 100 Kbps)
- Serial EEPROM interface for initializing Configuration registers and 8051 firmware
- Testability – JTAG support
- 12-MHz oscillator with internal Phase-Locked Loop (PLL) multiplier
- 1.0V and 3.3V operating voltages

- 15 physical USB endpoints
  - Endpoint 0 (EP 0) for device control and status
  - Six Dedicated endpoints for register and PCI Express accesses – CSROUT, CSRIN, PCIOUT, PCIIN, STATIN, and RCIN
  - Up to eight Configurable endpoints – GPEP[3:0], which can operate as *Isochronous*, *Bulk*, or *Interrupt* endpoint types
- Diagnostic register allows forced USB errors
- Software-controlled disconnect allows re-enumeration
- Atomic operation to Set and Clear *Status* bits simplifies software
- Lead-Free and RoHS (Reduction of Hazardous Substances)-compliant
- 10 x 10 mm^2, 136-pin Dual-Row QFN package
- Typical power – 788 mW
- Compliant to the following specifications:
  - *PCI Local Bus Specification, Revision 3.0 (PCI r3.0)*
  - *PCI Bus Power Management Interface Specification, Revision 1.2 (PCI Power Mgmt. r1.2)*
  - *PCI Code and ID Assignment Specification, Revision 1.2*
  - *PCI to PCI Bridge Architecture Specification, Revision 1.2 (P-to-P Bridge r1.1)*
  - *PCI Express Base Specification, Revision 1.1 (PCI Express Base r1.1)*
  - *PCI Express Base Specification, Revision 2.0 (PCI Express Base r2.0)*
  - *PCI Express Base Specification, Revision 2.0 Errata*
  - *PCI Express Base Specification, Revision 2.1 (PCI Express Base r2.1)*
  - *PCI Express Card Electromechanical Specification, Revision 2.0 (PCI ExpressCard CEM r2.0)*
  - *PCI Express Mini Card Electromechanical Specification, Revision 1.1 (PCI ExpressCard Mini CEM r1.1)*
  - *IEEE Standard 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture (IEEE Standard 1149.1-1990)*
  - *IEEE Standard 1149.1a-1993, IEEE Standard Test Access Port and Boundary-Scan Architecture*
  - *IEEE Standard 1149.1-1994, Specifications for Vendor-Specific Extensions*
  - *IEEE Standard 1149.6-2003, IEEE Standard Test Access Port and Boundary-Scan Architecture Extensions (IEEE Standard 1149.6-2003)*
  - *The I^2C-Bus Specification, Version 2.1 (I^2C Bus v2.1)*
  - *System Management Bus (SMBus) Specification, Version 2.0 (SMBus v2.0)*
  - *Universal Serial Bus Specification Revision 2.0 (USB r2.0)*
  - *Universal Serial Bus Specification Revision 3.0 (USB r3.0)*

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 2 Signal Pin Descriptions

2.1 Introduction

This chapter provides descriptions of the 136 USB 3382 signal pins and center Ground pad, which include the signal name, type, location, and a brief description. A map of the USB 3382's physical pin locations is also provided.

2.2 Pin Description Abbreviations

The following abbreviations are used in the signal pin tables provided in this chapter.

Table 2-1. Pin Assignment Abbreviations

Abbreviation	Description
#	Active-Low signal
A	Analog Input signal
APWR	3.3V Power for SerDes Analog circuits
CMLCLKn[a]	Differential low-voltage, high-speed, CML negative Clock inputs
CMLCLKp	Differential low-voltage, high-speed, CML positive Clock inputs
CMLRn	Differential low-voltage, high-speed, CML negative Receiver inputs
CMLRp	Differential low-voltage, high-speed, CML positive Receiver inputs
CMLTn	Differential low-voltage, high-speed, CML negative Transmitter outputs
CMLTp	Differential low-voltage, high-speed, CML positive Transmitter outputs
CPWR	1.0V Power for low-voltage Core circuits
DPWR	1.0V Power for SerDes Digital circuits
GND	Common Ground for all circuits
HCSLOUT	High-Speed Current Steering Logic (HCSL) Output clocks
I	Input
I/O	Bidirectional (Input or Output)
I/OPWR	3.3V Power for Input and Output interfaces
N/C	No Connect, these signals must not be connected to board electrical paths
O	Output
OD	Open Drain output
PLLPWR	1.0V Power pins for Phase-Locked Loop (PLL) circuits
PD	Weak internal pull-down resistor
PU	Weak internal pull-up resistor
PWR	3.3V power for the USB interface
SerDes	Serializer/De-Serializer differential low-voltage, high-speed, I/O signal pairs (negative and positive)
STRAP	Input signals used for USB 3382 configuration, operational mode Setting, and ***Factory Test***; these signals generally are not toggled at runtime

a. *For REFCLK input, CML source is recommended; however, LVDS source is supported.*

2.3 Internal Pull-Up/Pull-Down Resistors

The USB 3382 contains I/O buffers that have weak internal pull-up or pull-down resistors, indicated in this chapter by PU or PD, respectively, in the signal pin tables (**Type** column). If a signal with this notation is used and no board trace is connected to the pin, the internal resistor is usually sufficient to keep the signal from toggling. However, if a signal with this notation is not used, but is connected to a board trace and is not used nor driven by an external source at all times, the internal resistors might not be sufficiently strong, to hold the signal in the Inactive state. In cases such as these, it is recommended that the signal be pulled High to VDD_IO or Low to Ground, as appropriate, through a 3KΩ to 10KΩ resistor.

Table 2-2 lists the internal pull-up and pull-down resistor values.

Table 2-2. Internal Resistor Values

Internal Resistor	Minimum	Typical	Maximum	Units
PU	33.6K	50K	69.3K	Ω
PD	33.5K	50K	69.4K	Ω

2.4 Signal Pin Descriptions

Note: *If there is more than one pin per signal name that includes a numbered range, the locations are listed in the same sequence in which the range is listed, starting at the top row, from left to right. For example, PEX_PERn1 is located at B10, and PEX_PERN0 is located at A6.*

If there is more than one pin per signal name that does not include a numbered range (such as VDD_IO), the locations are listed in ascending alphanumeric order.

The USB 3382 signals are divided into the following groups:

- PCI Express Signals
- USB Interface Signals
- Serial EEPROM Signals
- Strapping Signals
- JTAG Interface Signals
- I^{2}C/SMBus Slave Interface Signals
- Device-Specific Signals
- No Connect Signals
- Power and Ground Signals

2.4.1 PCI Express Signals

Table 2-3 defines the PCI Express SerDes and Control signals.

Table 2-3. PCI Express Signals – 17 Pins

Signal Name	Type	Location	Description
PEX_PERn[1:0]	CMLRn	B10, A6	**Negative Half of PCI Express Receiver Differential Signal Pairs (2 Pins)**
PEX_PERp[1:0]	CMLRp	A11, A5	**Positive Half of PCI Express Receiver Differential Signal Pairs (2 Pins)**
PEX_PERST#	I PU	B15	**PCI Express Reset** When asserted, causes a full-device (Fundamental) reset. Must be asserted for 100 ms after power and clocks are stable. When operating in systems that do not implement the D3cold state, PEX_PERST# should be driven together with PWRON_RST# input. (Refer to Chapter 4, “Reset and Initialization,” for details.)
PEX_PETn[1:0]	CMLTn	A14, A8	**Negative Half of PCI Express Transmitter Differential Signal Pairs (2 Pins)**
PEX_PETp[1:0]	CMLTp	B11, B6	**Positive Half of PCI Express Transmitter Differential Signal Pairs (2 Pins)**
PEX_REFCLKn	CMLCLKn	B24	**Negative Half of 100-MHz PCI Express Reference Clock Input Signal Pair** PEX_REFCLKn/p do not require AC coupling capacitors, when driven from an HCSL source. Use with other Clock driver types (such as LVDS or Low-Voltage Positive Emitter-Coupled Logic (LVPECL)) has not been characterized.
PEX_REFCLKp	CMLCLKp	A27	**Positive Half of 100-MHz PCI Express Reference Clock Input Signal Pair** PEX_REFCLKn/p do not require AC coupling capacitors, when driven from an HCSL source. Use with other Clock driver types (such as LVDS or LVPECL) has not been characterized.
PEX_REFCLK_OUT_BIAS	A	B22	**Optional Bias Voltage Input** Can be left unconnected for typical applications. Suggest routing this pin to a test point.
PEX_REFCLK_OUTn2 PEX_REFCLK_OUTn1	HCSLOUT	A23 B21	**Negative Half of 100-MHz PCI Express Reference Clock Output for Pairs 2 and 1 (2 Pins)** Reference Clock outputs are enabled, by default, and can be disabled by Clearing their **Clock Enable** register *REFCLK x Enable* bit (Port 0, offset 1D8h[9:8]). ***Note:*** *Termination resistor networks are not required for REFCLK output pairs.*
PEX_REFCLK_OUTp2 PEX_REFCLK_OUTp1	HCSLOUT	A22 A24	**Positive Half of 100-MHz PCI Express Reference Clock Output for Pairs 2 and 1 (2 Pins)** Reference Clock outputs are enabled, by default, and can be disabled by Clearing their **Clock Enable** register *REFCLK x Enable* bit (Port 0, offset 1D8h[9:8]). ***Note:*** *Termination resistor networks are not required for REFCLK output pairs.*
PEX_REFCLK_OUT_RREF	A	B23	**External Reference Resistor** Connect to Ground through a 2.00KΩ, 1% resistor. Place the resistor close to this pin.

2.4.2 USB Interface Signals

The USB 3382 includes signals for interfacing to a USB Host, defined in Table 2-4. For information regarding USB Interface use, refer to Chapter 8, "USB Controller Functional Description."

Table 2-4. USB Interface Signals – 8 Pins

Signal Name	Type	Location	Description
USB_DM	I/O	A3	**High-Speed USB Negative Data Port**
USB_DP	I/O	A4	**High-Speed USB Positive Data Port**
USB_RREF	A	A2	**External Reference Resistor** Connect USB_RREF to USB_AVSS through a 1.6KΩ ±1% resistor. Place the resistor close to the signal pad. Keep traces as short as possible.
USB_RXM	CMLRn	A61	**SuperSpeed USB Receive Differential Data, Negative Half**
USB_RXP	CMLRp	A62	**SuperSpeed USB Receive Differential Data, Positive Half**
USB_TXM	CMLTn	A59	**SuperSpeed USB Transmit Differential Data, Negative Half**
USB_TXP	CMLTp	A60	**SuperSpeed USB Transmit Differential Data, Positive Half**
USB_VBUS	I	C1	**VBUS Presence Detect** USB_VBUS is used to sense when a USB Host is connected. Connect to the VBUS pin of the USB cable receptacle, through a 27KΩ resistor. Also, connect to Ground through a 47KΩ resistor. The resistor network divides the 5V at the USB receptacle's VBUS pin, so that it is in the 3V range at the USB 3382's USB_VBUS input. ***Note:*** *When the USB 3382 is not powered (VDD_IO = 0V), the USB_VBUS pin presents a leakage path to Ground when an external USB Host is connected with the connector VBUS pin at 5V. This leakage current is limited by the external 27KΩ resistor to approximately 150 µA. For designs that require zero leakage when the USB 3382 is not powered, an external FET is recommended.*

2.4.3 Serial EEPROM Signals

The USB 3382 includes four signals for interfacing to a serial EEPROM, defined in Table 2-5. For information regarding serial EEPROM use, refer to Chapter 5, "Serial EEPROM Controller."

Table 2-5. Serial EEPROM Signals – 4 Pins

Signal Name	Type	Location	Description
EE_CS#	I/O PU	B32	**Active-Low Serial EEPROM Chip Select Output** ***Note:*** *Although this is an I/O signal, its logical operation is output.*
EE_WRDATA/EE_DI	O	A36	**USB 3382 Output to Serial EEPROM Data Input** Used for writing Serial data when programming the serial EEPROM.
EE_RDDATA/EE_DO	I/O PU	B33	**USB 3382 Input from Serial EEPROM Data Output** Inputs data from the serial EEPROM during Read operations. Should be pulled High to VDD_IO. ***Note:*** *Although this is an I/O signal, its logical operation is input.*
EE_CLK/EE_SK	I/O PU	A38	**Serial EEPROM Clock Frequency Output** Programmable, by way of the **Serial EEPROM Clock Frequency** register *EepFreq[2:0]* field (Port 0, offset 268h[2:0]), to the following: • 1 MHz (default) • 1.98 MHz • 5 MHz • 9.62 MHz • 12.5 MHz • 15.6 MHz • 17.86 MHz ***Note:*** *Although this is an I/O signal, its logical operation is output.*

2.4.4 Strapping Signals

The USB 3382 Strapping inputs, defined in Table 2-6, Set the configuration of Link width, Spread-Spectrum clocking, and various setup and test modes. These inputs must be pulled High to VDD_IO or Low to Ground, or left unconnected, as indicated in the table.

After a Fundamental Reset, the **Link Capability** (All Ports, offset 74h), and **Debug Control** and **Port Configuration** registers (Port 0, offsets 1DCh and 574h, respectively) capture pin status. Strapping input Configuration data can be changed, by writing new data to these registers from the serial EEPROM. I²C/SMBus can also change Strapping input Configuration data; however, it should first Set the Port/SuperSpeed USB's **Port and SuperSpeed USB Control** register *Disable Port x/SuperSpeed USB* bit (Port 0, offset 234h[19:16]), to prevent linkup and Host enumeration. Then, when I²C programming is complete, I²C/SMBus should lastly Clear Port 0's *Disable Port x* bit, to enable linkup and allow subsequent Host enumeration.

Table 2-6. Strapping Signals – 15 Pins

Signal Name	Type	Location	Description
STRAP_DEBUG_SEL#	I PU	C4	***Factory Test Only*** STRAP_DEBUG_SEL# can be left unconnected in standard applications. If this input is connected to a board circuit trace, it must be externally pulled High to VDD_IO.
STRAP_FAST_BRINGUP#	I PU	A40	***Factory Test Only*** STRAP_FAST_BRINGUP# can be left unconnected in standard applications. If this input is connected to a board circuit trace, it must be externally pulled High to VDD_IO.
STRAP_LEGACY	I PU	C13	**USB Legacy Mode Select** When STRAP_LEGACY is pulled High to VDD_IO, the USB 3382's USB endpoints and register set are backward-compatible for applications that run on the NET 2282 USB device Controller. When STRAP_LEGACY is pulled or tied to Ground, the USB 3382 operates with advanced registers and endpoints. (Refer to Chapter 7, "PCI Express Interface," and Chapter 8, "USB Controller Functional Description," for further details.)
STRAP_PLL_BYPASS#	I PU	C12	***Factory Test Only*** STRAP_PLL_BYPASS# can be left unconnected in standard applications. If this input is connected to a board circuit trace, it must be externally pulled High to VDD_IO.
STRAP_PORTCFG	I PD	B37	**Port Configuration Select** Used to select the USB 3382's Port configuration. The STRAP_PORTCFG value is reflected in the initial **Port Configuration** register *Port Configuration* bit (Port 0, offset 574h[0]) value. This configuration Setting can be subsequently overwritten by serial EEPROM load. (Refer to Section 7.3, "Enhanced Adapter Mode," and Section 7.4, "Root Complex Mode," for details.) L = x1, x1 (each Port operates at Gen 1 or Gen 2 Link rate) H = x2 (Gen 1 Link rate only) ***Note:*** *For the x1x1 Port configuration, downstream Port 1 is available only when STRAP_LEGACY=L.*

Table 2-6. Strapping Signals – 15 Pins *(Cont.)*

<table>
<tr><th>Signal Name</th><th>Type</th><th>Location</th><th>Description</th></tr>
<tr><td>STRAP_PROBE_MODE#</td><td>I
PU</td><td>C3</td><td>Factory Test Only
STRAP_PROBE_MODE# can be left unconnected in standard applications. If this input is connected to a board circuit trace, it must be externally pulled High to VDD_IO.</td></tr>
<tr><td>STRAP_RC_MODE</td><td>I
PD</td><td>B35</td><td>Root Complex Mode Enable

Root Complex Mode
When STRAP_RC_MODE is pulled High to VDD_IO, the USB 3382 operates as a PCI Express Root Complex. In this mode, PCI Express Lanes 0 and 1 function as downstream Ports.
(Refer to Chapter 7, “PCI Express Interface,” for further details.)

Adapter Mode
When STRAP_RC_MODE is pulled or tied Low to Ground, the USB 3382 operates as a PCI Express Adapter (endpoint). In this mode, the PCI Express interface functions as an upstream Port.
(Refer to Chapter 7, “PCI Express Interface,” and Chapter 8, “USB Controller Functional Description,” for further details.)</td></tr>
<tr><td>STRAP_SERDES_MODE_EN#</td><td>I
PU</td><td>A39</td><td>Factory Test Only
STRAP_SERDES_MODE_EN# can be left unconnected in standard applications. If this input is connected to a board circuit trace, it must be externally pulled High to VDD_IO.</td></tr>
<tr><td>STRAP_SMBUS_EN#</td><td>I
PU</td><td>B28</td><td>System Management Bus Enable

Selects the I2C or SMBus protocol, and defines the default SMBus Configuration register SMBus Enable bit (Port 0, offset 2ACh[0]) value, as listed below.

<table><tr><th>STRAP_SMBUS_EN#
Input State</th><th>Offset 2ACh[0]
Value</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>
When pulled or tied Low to Ground, enables SMBus Slave protocol on the I2C_SCL and I2C_SDA 2-wire bus. ARP is enabled if I2C_ADDR2 is Low when sampled at the time PEX_PERST# input de-asserts.

When pulled High to VDD_IO, enables I2C Slave protocol on the I2C_SCL and I2C_SDA 2-wire bus.</td></tr>
<tr><td>STRAP_SSC_CENTER#</td><td>I
PU</td><td>C11</td><td>SuperSpeed USB Center-Spread Clock
When STRAP_SSC_CENTER# is pulled or tied Low to VSS (Ground), the USB 3382’s USB interface can support a USB Host or hub that uses a Center-Spread Reference Clock. Otherwise, this signal can be left unconnected.</td></tr>
</table>

Table 2-6. Strapping Signals – 15 Pins *(Cont.)*

Signal Name	Type	Location	Description
STRAP_TESTMODE[3:0]	[3, 1]: I PU [2, 0]: I PD	C10, C9, C8, C7	***Factory Test Only* (4 Pins)** STRAP_TESTMODE[3:0] select the USB 3382 clocking configuration. Supported strapping combinations are as follows: • Dual Clocks configuration = HLLL – PCI Express interface is clocked from PEX_REFCLKn/p input (100 MHz) – PEX_REFCLK_OUTn*x*/p*x* (100 MHz) are buffered from PEX_REFCLKn/p input – USB interface is clocked from XTAL_IN input (30 MHz) • 30 MHz Only configuration = HLLH – USB and PCI Express interfaces are both clocked from XTAL_IN (30 MHz) – PEX_REFCLK_OUTn*x*/p*x* signals (100 MHz) are generated from XTAL_IN (30 MHz) All other strapping combinations are ***Factory Test Only***: H = Pull High to VDD_IO L = Pull or tie Low to VSS (Ground)
STRAP_UPCFG_TIMER_EN#	I PU	A28	**Link Upconfigure Timer Enable** STRAP_UPCFG_TIMER_EN# maps to the **Debug Control** register *STRAP_UPCFG_TIMER_EN# Pin State* bit (Port 0, offset 1DCh[4]). This input and its corresponding register bit must ***not*** be toggled at runtime. When STRAP_UPCFG_TIMER_EN# is pulled High, the Data Rate Identifier symbol in the TS Ordered-Sets always advertises support for both the 5.0 GT/s (Gen 2) data rate and Autonomous Change. When STRAP_UPCFG_TIMER_EN# is pulled or tied Low, if this Link training sequence fails during the *Configuration* state, the next time the LTSSM exits the *Detect* state, TS Ordered-Sets advertise only the 2.5 GT/s (Gen 1) data rate and no Autonomous Change support. If Link training continues to fail when the LTSSM is in the *Configuration* state, the LTSSM continues to alternate between Gen 1 and Gen 2 advertisement every time it exits the *Detect* state. ***Note:*** *This feature should only be enabled if a non-compliant device will not linkup when these Data Rate Identifier bits are Set.*

2.4.5 JTAG Interface Signals

The USB 3382 includes five signals for performing Joint Test Action Group (JTAG) boundary scan, defined in Table 2-7. The JTAG interface is described in Section 17.7, "JTAG Interface."

Table 2-7. JTAG Interface Signals – 5 Pins

Signal Name	Type	Location	Description
JTAG_TCK	I PU	A48	**JTAG Test Clock Input** JTAG Test Access Port (TAP) Controller clock source. Frequency can be from 0 to 20 MHz.
JTAG_TDI	I PU	A47	**JTAG Test Data Input** Serial input to the JTAG TAP Controller, for test instructions and data.
JTAG_TDO	O	B29	**JTAG Test Data Output** Serial output from the JTAG TAP Controller test instructions and data.
JTAG_TMS	I PU	B42	**JTAG Test Mode Select** Input decoded by the JTAG TAP Controller, to control test operations.
JTAG_TRST#	I PU	A34	**JTAG Test Reset** Active-Low input used to reset the Test Access Port. When JTAG functionality is not used, the JTAG_TRST# input should be driven Low, or pulled Low to VSS (Ground) through a 1.5KΩ resistor, to place the JTAG TAP Controller into the *Test-Logic-Reset* state, which disables the test logic and enables standard logic operation. Alternatively, if JTAG_TRST# input is High, the JTAG TAP Controller can be placed into the *Test-Logic-Reset* state by initializing the JTAG TAP Controller's **Instruction** register to contain the *IDCODE* instruction, or by holding the JTAG_TMS input High for at least five rising edges of the JTAG_TCK input.

2.4.6 I^{2}C/SMBus Slave Interface Signals

Table 2-8 defines the five I^{2}C/SMBus Slave interface signals. The I^{2}C/SMBus Slave interface provides access to the USB 3382 registers. Most registers that can be programmed by serial EEPROM (including registers that are Read Only to PCI Express), can be also programmed by I^{2}C/SMBus. However, I^{2}C/SMBus cannot replace the serial EEPROM programming of some registers, *such as* SerDes--related registers that (generally) must be configured prior to automatic link training, immediately following Reset. I^{2}C/SMBus can also be used to program the serial EEPROM.

The I^{2}C/SMBus Slave interface used with PLX software is a powerful debugging tool that enables register access, regardless of whether the PCI Express Link is Up. Figure 2-1 illustrates the suggested Header pin layout on 0.1-inch centers, for direct connection to the Total Phase Aardvark I^{2}C-USB converter (as used with PLX Rapid Development Kits (RDKs)).

For further details, refer to Chapter 12, "I^{2}C/SMBus Slave Interface Operation."

Figure 2-1. Suggested I^{2}C Header Pin Layout on 0.1-Inch Centers

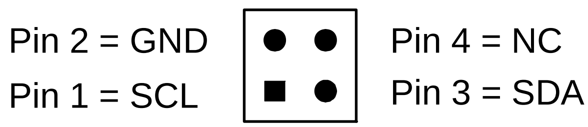


Table 2-8. I^2C/SMBus Slave Interface Signals – 5 Pins

<table>
<tr><th>Signal Name</th><th>Type</th><th>Location</th><th>Description</th></tr>
<tr><td>I2C_ADDR[2:0]</td><td>I
PU</td><td>A41, B36, A42</td><td>

I^2C Slave/SMBus Device Address Bits 2 through 0 Inputs (3 Pins)

Used to define the three least significant bits of the USB 3382 7-bit I^2C Slave/SMBus Device Address, which is reflected in both the **I^2C Configuration** register *Slave Address* and **SMBus Configuration** register *SMBus Device Address* fields (Port 0, offsets 294h[2:0] and 2ACh[7:1], respectively). If I^2C or SMBus configuration is used, the I2C_ADDR[2:0] inputs should be strapped to a unique address, to avoid an address conflict with any other I^2C/SMBus devices (on the same I^2C Bus/SMBus segment).

The following table lists the pin and register bit values associated with the I^2C Slave and SMBus Device addresses.

I2C_ADDR [2:0] Values	Offset 294h[2:0] Value	Offset 294h[6:3] Value	I^2C Slave Address	SMBus Device Address
000	000b	1011b	1011_000b	ARP
001	001b		1011_001b	ARP
010	010b		1011_010b	ARP
011	011b		1011_011b	ARP
100	100b		1011_100b	1011_100b
101	101b		1011_101b	1011_101b
110	110b		1011_110b	1011_110b
111	111b		1011_111b	1011_111b

If the STRAP_SMBUS_EN# input (of which its inverse value defines the default value of the **SMBus Configuration** register *SMBus Enable* bit (Port 0, offset 2ACh[0])) is Low, to enable SMBus protocol as default, the I2C_ADDR2 input defines the same default value for two register bits – bit 2 of the Slave address (Port 0, offset 294h[2]), and the **SMBus Configuration** register *ARP Disable* bit (Port 0, offset 2ACh[8]). Specifically, if the STRAP_SMBUS_EN# input is Low (to enable SMBus protocol as default):

- If the I2C_ADDR2 input is Low, to enable Address Resolution Protocol (ARP) as default, the *ARP Disable* bit default value is 0, and bit 2 of the **I^2C Configuration** register *Slave Address* field defaults to a value of 0. In this configuration, the upper five bits of the 7-bit Slave address default to value 10110b.
- If the I2C_ADDR2 input is High, to disable ARP as default, the *ARP Disable* bit default value is 1, and bit 2 of the **I^2C Configuration** register *Slave Address* field defaults to a value of 1. In this configuration, the upper five bits of the 7-bit Slave address default to value 10111b.

The internal pull-up resistors cause the I2C_ADDR[2:0] inputs to default to HHH (111b). If the I^2C/SMBus Slave address must be changed to a different value, add pull-down resistors, to force the appropriate I2C_ADDR[2:0] inputs to a logic Low state, and the Address bits that are to be logic High can be pulled High, or optionally, can remain unconnected. However, if the I2C_ADDR[2:0] inputs that are logic High are connected to circuit traces, external pull-up resistors are recommended, because the internal pull-up resistors might not be sufficiently strong, to hold the input(s) High.

</td></tr>
</table>

Table 2-8. I^2C/SMBus Slave Interface Signals – 5 Pins *(Cont.)*

Signal Name	Type	Location	Description
I2C_SCL	OD	A45	**I^2C/SMBus Serial Clock Line** I^2C/SMBus Clock line. Data on the I^2C Bus can be transferred at rates of up to 100 kbit/s (Standard mode). **I2C_SCL requires an external pull-up resistor.** ***Note:*** *The USB 3382 I^2C/SMBus Slave Interface can stretch the Low period of the I^2C/SMBus clock while a simultaneous in-band Request that also targets USB 3382 registers is being processed.*
I2C_SDA	OD	B39	**I^2C/SMBus Serial Data I/O** Transmits and receives I^2C/SMBus data during I^2C/SMBus accesses to USB 3382 registers. **I2C_SDA requires an external pull-up resistor.**

2.4.7 Device-Specific Signals

Table 2-9 defines the Device-Specific signals – signals that are unique to the USB 3382.

Table 2-9. Device-Specific Signals – 18 Pins

<table>
<tr><th>Signal Name</th><th>Type</th><th>Location</th><th>Description</th></tr>
<tr><td>CPU_RXD</td><td>I</td><td>B38</td><td>8051 Serial Port Receive Data Input</td></tr>
<tr><td>CPU_TXD</td><td>O</td><td>A44</td><td>8051 Serial Port Transmit Data Output</td></tr>
<tr><td>FATAL_ERR#</td><td>O</td><td>B40</td><td>Fatal Error Output
Asserted Low when a Fatal error is detected in the USB 3382 and the following conditions exist (all the same conditions that are required to send a Fatal Error Message to the Host).
For PCI Express Base r2.1 errors:
• Specific error is defined as Fatal in the Uncorrectable Error Severity register (All Ports and USB Controller, offset FC0h), and
• Reporting of the specific error condition is enabled (not masked) by the Uncorrectable Error Mask register's (All Ports and USB Controller, offset FBCh) corresponding Interrupt Mask bit, and
• Device Control register Fatal Error Reporting Enable bit (All Ports and USB Controller, offset 70h[2]) –or– PCI Command register SERR# Enable bit (All Ports, offset 04h[8]) is Set
The Device Status register Fatal Error Detected bit (All Ports and USB Controller, offset 70h[18]) is Set, and the specific error is flagged in the Uncorrectable Error Status register (All Ports and USB Controller, offset FB8h).
If FATAL_ERR# output is not used, this signal can remain unconnected.</td></tr>
<tr><td>GPIO[3:2]</td><td>I/O
PU</td><td>B16, B55</td><td>General-Purpose I/O (2 Pins)
General-Purpose input/output (I/O) signals providing input, output, or pulse-width-modulated (PWM) output functions for specific applications. (Refer to Chapter 11, "GPIO Controller Functional Description," for further details.)</td></tr>
<tr><td>LANE_GOOD[1:0]#</td><td>I/O
PU</td><td>A16, A19</td><td>General-Purpose I/O [1:0] (Default) or Active-Low PCI Express Lane Status Indicator Outputs for Lanes [1-0] (2 Pins)
Default function for these pins is GPIO[1:0], as determined by the Debug Control register LANE_GOODx#/GPIOx Pin Function Select bit (Port 0, offset 1DCh[22], is Cleared). GPIO[1:0] provide input, output, and/or pulse-width-modulated (PWM) output functions for specific applications.
Alternately, LANE_GOOD[1:0]# can be programmed to indicate PCI Express Link status for the two PCI Express Ports, and can directly drive the common-anode LED module. LED behavior when connected to LANE_GOOD[1:0]#:
• Solid Off – Lane is disabled
• Solid On – Lane is enabled, 5.0 GT/s
• 0.5 seconds On, 0.5 seconds Off – Lane is enabled, 2.5 GT/s
(Refer to Chapter 11, "GPIO Controller Functional Description," and Section 17.8, "Lane Good Status LEDs," for further details.)</td></tr>
</table>

Table 2-9. Device-Specific Signals – 18 Pins *(Cont.)*

Signal Name	Type	Location	Description
MFG_AMC	I PD	C5	***Factory Test Only*** Do not connect this pin to board electrical paths.
MFG_TAPEN	I PD	C6	***Factory Test Only*** Do not connect this pin to board electrical paths.
MFG_TMC1	I PD	C15	***Factory Test Only*** Do not connect this pin to board electrical paths.
MFG_TMC2	I PD	C16	***Factory Test Only*** Do not connect this pin to board electrical paths.
PEX_INTA#	OD	B34	**Interrupt Output** PEX_INTA# Interrupt output is enabled if: • INT*x* Messages are enabled (**PCI Command** register *Interrupt Disable* bit (All Ports, offset 04h[10]) is Cleared), and MSIs are disabled (**MSI Control** register *MSI Enable* bit (All Ports and USB Controller, offset 48h[16]) is Cleared) • PEX_INTA# output is enabled (**ECC Error Check Disable** register *Enable PEX_INTA# Interrupt Output(s) for GPIO-Generated Interrupts* bit (All Ports and USB Controller, offset 1C8h[6]) is Set) The three interrupt mechanisms, listed below, are mutually exclusive modes of operation, on a per-Port basis, for all interrupt sources: • Conventional PCI INT*x* Message generation • Native MSI transaction generation • Device-Specific PEX_INTA# assertion PEX_INTA# assertion (Low) indicates that one or more of the following events and/or errors (if not masked) were detected: • USB events • DMA events • General-Purpose Input Interrupt events (Refer to Chapter 13, “Interrupt and Status Register Operation,” for details.) **If used, PEX_INTA# requires an external pull-up to VDD_IO.**
PROCMON	O	C14	***Factory Test Only*** Do not connect this pin to board electrical paths.
PWRON_RST#	I PU	B14	**Power-On Reset** When operating in systems in which remote Wakeup, using WAKE# or beacon signaling from the D3cold state, must be forwarded, PWRON_RST# must be held High when the USB 3382 is in the D3cold state. When operating in systems that do not implement the D3cold state, PWRON_RST# should be driven together with PEX_PERST#. (Refer to Chapter 4, “Reset and Initialization,” for details.)

Table 2-9. Device-Specific Signals – 18 Pins *(Cont.)*

Signal Name	Type	Location	Description
WAKE#	OD	B56	**PCI Express WAKE#** Bidirectional signal used in systems that implement the remote Wakeup function. When the USB 3382 is in the D3cold state: • USB 3382 asserts WAKE# Low, if a PCI Express beacon is received by a downstream Port • If another device asserts WAKE# Low, the USB 3382 drives PCI Express beacon signaling toward the Root Complex When operating in systems that do not implement the D3cold state, PWRON_RST# should be driven together with PEX_PERST#. WAKE# should be externally pulled High to VAUX_IO.
XTAL_IN	I/O PU	B30	**External Crystal/Oscillator Input** Connect to a 30-MHz external crystal resonator circuit or a CMOS oscillator. (Refer to the USB 3382 RDK reference schematic for an example.) When selecting a crystal, specify as follows: • **Nominal Frequency** – 30.000 MHz • **Cut** – AT Fundamental • **Frequency Tolerance** – ±100 ppm • **Stability over Temperature** – ±100 ppm • **Load Capacitance** – 16 to 18 pF ***Note:*** *Although this is an I/O signal, its logical operation is input.*
XTAL_OUT	O	A35	**External Crystal Output** If a crystal resonator circuit is used, connect this output to the crystal. If an external oscillator is used to drive XTAL_IN, leave this output unconnected.

2.4.8 No Connect Signals

Notice: ***Do not connect these pins to board electrical paths. These pins are internally connected to the device.***

Table 2-10. No Connect Signals – 11 Pins

Signal Name	Type	Location	Description
N/C	***Reserved***	A1, A17, A20, A21, A33, A49, A52, A53, A55, B25, B47	**No Connect (11 Pins)** Do not connect these pins to board electrical paths.

2.4.9 Power and Ground Signals

Table 2-11. Power and Ground Signals – 48 Pins and Center Pad

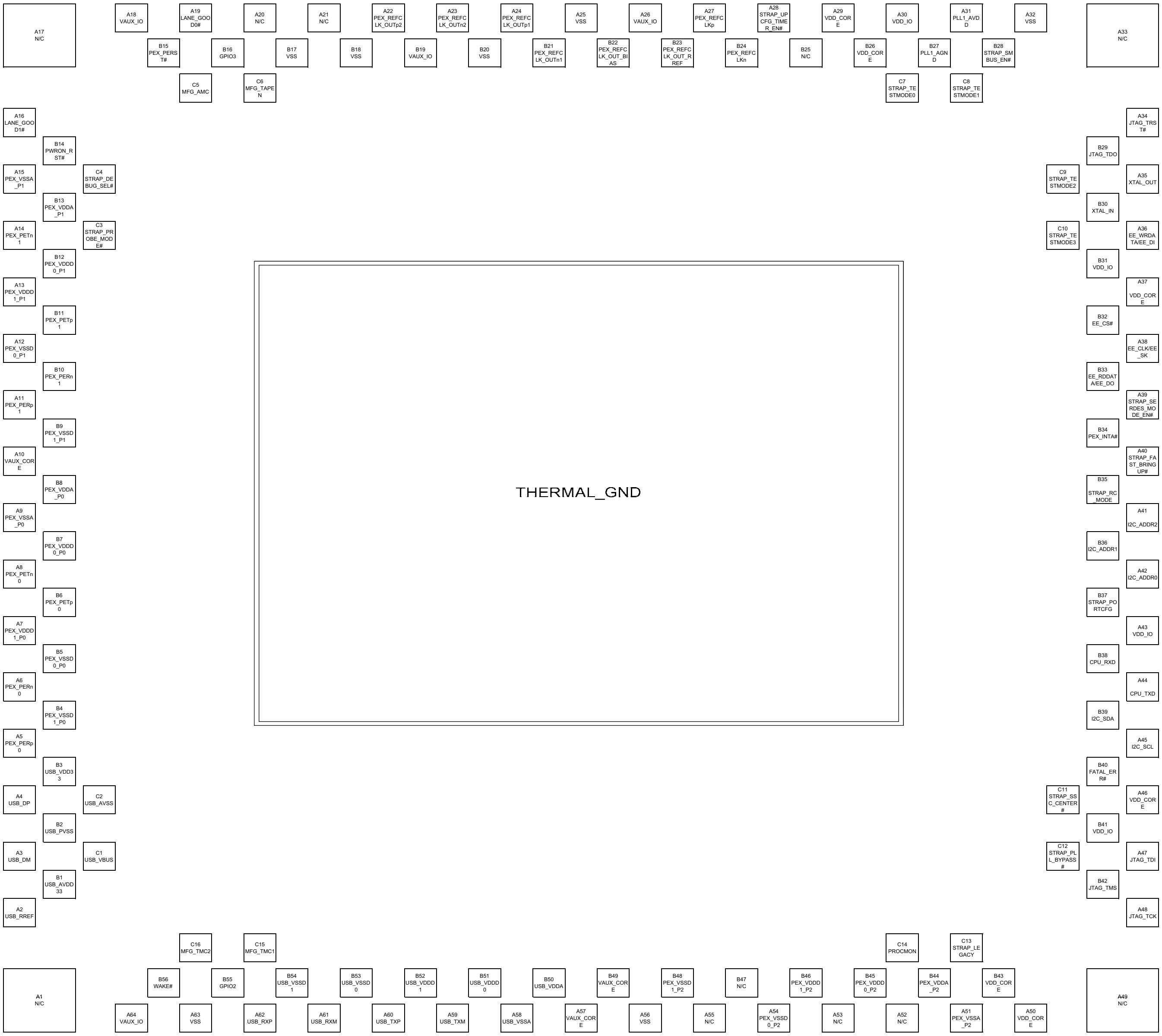
Signal Name	Type	Location	Description
PEX_VDDA_P2 PEX_VDDA_P1 PEX_VDDA_P0	APWR	B44 B13 B8	**3.3V SerDes Analog Power Supply for PCI Express Interface (3 Pins)** Must be the same voltage as VDD_IO. ***Note:*** *If stand-by power is implemented, power for these pins should be derived from the Vaux supply.*
PEX_VDDD0_P2 PEX_VDDD0_P1 PEX_VDDD0_P0	DPWR	B45 B12 B7	**1.0V SerDes Digital Power Supply for PCI Express Interface (3 Pins)** ***Note:*** *If stand-by power is implemented, power for these pins should be derived from the Vaux supply.*
PEX_VDDD1_P2 PEX_VDDD1_P1 PEX_VDDD1_P0	DPWR	B46 A13 A7	**1.0V SerDes Digital Power Supply for PCI Express Interface (3 Pins)** ***Note:*** *If stand-by power is implemented, power for these pins should be derived from the Vaux supply.*
PEX_VSSA_P2 PEX_VSSA_P1 PEX_VSSA_P0	GND	A51 A15 A9	**SerDes Analog Ground for PCI Express Interface (3 Pins)** Connect to VSS (Ground).
PEX_VSSD0_P2 PEX_VSSD0_P1 PEX_VSSD0_P0	GND	A54 A12 B5	**SerDes Digital Ground for PCI Express Interface (3 Pins)** Connect to VSS (Ground).
PEX_VSSD1_P2 PEX_VSSD1_P1 PEX_VSSD1_P0	GND	B48 B9 B4	**SerDes Digital Ground for PCI Express Interface (3 Pins)** Connect to VSS (Ground).
PLL_AGND	GND	B27	**PLL Analog Ground** Connect to VSS (Ground).
PLL_AVDD	PLLPWR	A31	**1.0V Analog Power for PLL Circuits** Connect to the main 1.0V supply (VDD_CORE) through an Inductor Capacitor (LC) filter circuit.
THERMAL_GND	GND	CENTER PAD	**Ground**
USB_AVDD33	APWR	B1	**3.3V Analog Power for USB Interface** ***Note:*** *If stand-by power is implemented, power for these pins should be derived from the Vaux supply.*
USB_AVSS	GND	C2	**Connect to Ground (VSS)** Analog Ground for USB interface.
USB_PVSS	GND	B2	**Connect to Ground (VSS)** Digital Ground for USB interface.

Table 2-11. Power and Ground Signals – 48 Pins and Center Pad *(Cont.)*

Signal Name	Type	Location	Description
USB_VDD33	PWR	B3	**3.3V Digital Power for USB Interface** ***Note:*** *If standby power is implemented, power for this pin should be derived from the Vaux supply.*
USB_VDDA	APWR	B50	**3.3V SerDes Analog Power Supply for USB Interface** Must be the same voltage as VDD_IO. ***Note:*** *If standby power is implemented, power for this pin should be derived from the Vaux supply.*
USB_VDDD0	DPWR	B51	**1.0V SerDes Digital Power Supply for USB Interface** ***Note:*** *If standby power is implemented, power for this pin should be derived from the Vaux supply.*
USB_VDDD1	DPWR	B52	**1.0V SerDes Digital Power Supply for USB Interface** ***Note:*** *If standby power is implemented, power for this pin should be derived from the Vaux supply.*
USB_VSSA	GND	A58	**SerDes Analog Ground for USB Interface** Connect to VSS (Ground).
USB_VSSD0	GND	B53	**SerDes Digital Ground (VSS) for USB Interface** Connect to VSS (Ground).
USB_VSSD1	GND	B54	**SerDes Digital Ground (VSS) for USB Interface** Connect to VSS (Ground).
VAUX_CORE	CPWR	A10, A57, B49	**Auxiliary Core Voltage (3 Pins)** 1.0V, derived from Vaux. If standby power is not implemented, connect to VDD_CORE.
VAUX_IO	I/OPWR	A18, A26, A64, B19	**Auxiliary I/O Power (4 Pins)** 3.3V, derived from Vaux. If standby power is not implemented, connect to VDD_IO.
VDD_CORE	CPWR	A29, A37, A46, A50, B26, B43	**Core Logic Supply Voltage (6 Pins)** 0.95 to 1.10V.
VDD_IO	I/OPWR	A30, A43, B31, B41	**I/O Supply Voltage (4 Pins)** 3.3V.
VSS	GND	A25, A32, A56, A63, B17, B18, B20	**Ground (7 Pins)**

2.5 Physical Layout

Figure 2-2. USB 3382 Physical Signal Locations (Bottom View, as seen from Underside of Chip)



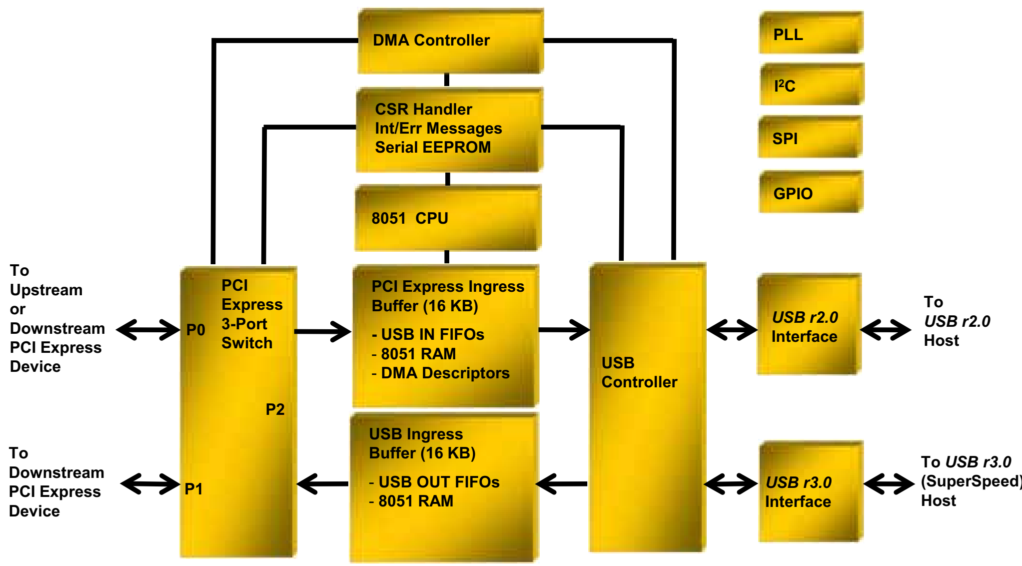
Note: *Drawing not to scale.*

Chapter 3 Functional Overview

3.1 Introduction

A functional block diagram of the USB 3382 is shown in Figure 3-1.

Figure 3-1. USB 3382 Block Diagram



3.1.1 Data Path

The data path for the bridge consists of three main parts:

- PCI Express interface
- USB Controller and *USB r2.0*, *USB r3.0* line interfaces
- Central RAM

From the PCI Express side, the USB 3382 provides two Lanes, each of which is capable of operating at Gen 2 (5.0 GT/s) data rates. The two PCI Express Lanes can be combined as single x2 Gen 1 Port, or they can be configured as two x1 Gen 1 or Gen 2 Ports (Ports 0 and 1), as illustrated in Figure 3-1. The external Ports are part of a three-Port PCI Express switch, the third Port of which (internal Port 2) connects to the USB Controller (PCI Express Adapter (endpoint)), and central RAM. When Port 0 is configured as an upstream Port, the USB 3382 can exist as a device in a USB Controller endpoint. When Port 0 is configured as a downstream Port, the USB 3382 functions as a PCI Express Root Complex, sitting at the top of a PCI Express hierarchy.

Note: *Port 2 is an internal virtual PCI-to-PCI bridge that connects the USB Controller to the PCI Express fabric.*

From the USB side, the USB 3382 presents a *USB r3.0* (SuperSpeed USB) client Port. The USB line interface consists of two different PHY blocks, one for *USB r2.0* (Full-Speed and High-Speed) signaling, and the other for *USB r3.0* (SuperSpeed) signaling. The USB Controller block connects the *USB r2.0*, *USB r3.0* blocks to the central RAM, as shown.

Central RAM provides a way-point for data moving from PCI Express to USB, and vice-versa. The PCI Express Ingress buffer receives data from the PCI Express interface. This includes data bound for USB IN endpoints, downstream PCI Express devices, and internal registers. The USB Ingress buffer receives data from the USB Controller received by way of an OUT endpoint. The central RAM also provides space for storing DMA Descriptors, as well as code and Data space for the integrated 8051 Micro-Controller.

PCI Express functions are described in Chapter 7, "PCI Express Interface." The USB Controller is described in Chapter 8, "USB Controller Functional Description." The *USB r2.0* and *USB r3.0* interfaces are described in Chapter 9, "USB r2.0 and USB r3.0 Functional Description."

3.1.2 DMA Controller

While the flow of data between the USB Controller and endpoint FIFOs in the central RAM is largely driven by the USB Host, the flow of data between the central RAM and PCI Express must either be explicitly handled by direct CPU accesses (by way of the PCI Express interface) or DMA transfers. The USB 3382 DMA Controller automates the flow of data between endpoint FIFOs in the central RAM and PCI Express interface, and relieves the local CPU of the task of moving bulk data.

The DMA Controller provides four independent channels – DMA Channels 0, 1, 2, and 3. Each DMA channel is associated with one pair of IN/OUT endpoints. For the IN endpoints, the DMA Controller issues Read Requests to the PCI Express interface. Completions from the PCI Express interface Reads are then written directly to the designated IN endpoint FIFO area within the PCI Ingress RAM. For an OUT endpoint, the DMA Controller reads Packet data from an OUT endpoint FIFO area within the USB Ingress RAM, and generates Memory Write TLPs to the PCI Express interface.

The DMA Controller can perform single transfers (Block mode), or it can execute a series of block transfers based upon a Descriptor list (Scatter Gather mode). For faster access, DMA Descriptors can be stored on-chip within the central RAM.

DMA Controller functions are described in Chapter 10, "DMA Controller."

3.1.3 8051 Micro-Controller Unit

The 8051 Micro Controller Unit (MCU), when used, enables the USB 3382 to function as a stand-alone System-on-Chip (SOC) processor. Use of the 8051 is mainly limited to special applications in which the USB 3382 is configured as a PCI Express Root Complex. In this mode, the 8051 can run from serial EEPROM-based firmware. This firmware can perform enumeration of downstream PCI Express devices, handle USB endpoints, interrupts, and service the DMA channels.

The 8051 is described in Chapter 6, "8051 Micro-Controller Unit."

3.1.4 General-Purpose Input/Output

The USB 3382 provides four General-Purpose Input/Output (GPIO) pins (signals). As inputs, GPIO pins can sense external signals and generate interrupts on High or Low transitions. As outputs, GPIO pins can output a constant value (High or Low), or they can output a programmable pulse-width-modulated (PWM) output.

GPIO functions are described in Chapter 11, "GPIO Controller Functional Description."

3.1.5 Reference Clock Outputs

The USB 3382 provides two pairs of PCI Express-compliant, buffered, 100-MHz, HCSL output clocks, PEX_REFCLK_OUTn*x*/p*x*, one pair for each of its downstream Ports. Clock outputs are buffered versions of the input Reference Clock pair, PEX_REFCLKn/p.

When operating as a PCI Express Root Complex (STRAP_RC_MODE=H), each of these pairs can be used to drive clocks to up to two downstream PCI Express devices. When operating as an adapter (STRAP_RC_MODE=L), one pair of REFCLK outputs can be used to drive a Reference Clock to a downstream PCI Express device on Port 1.

Each clock output pair can be disabled by software or serial EEPROM when not in use, for additional power savings, by Clearing their **Clock Enable** register *REFCLK x Enable* bit (Port 0, offset 1D8h[9:8])

It is recommended that system designs that make use of PEX_REFCLK_OUT outputs limit their total PC board trace lengths to approximately 10 inches or less, with one connector.

3.1.6 I²C/SMBus Slave Interface

The USB 3382 provides an I²C/SMBus Slave interface. This interface can be used to access internal registers and other debug features of the USB 3382, independently of the PCI Express and USB interfaces.

I²C/SMBus functions are described in Chapter 12, "I²C/SMBus Slave Interface Operation."

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 4 Reset and Initialization

4.1 Overview

The USB 3382 initialization sequence consists of the following:

1. The USB 3382 PEX_PERST# and PWRON_RST# inputs are asserted and de-asserted.
2. Serial EEPROM content is loaded into the Configuration registers and, optionally, the 8051 Program RAM.
3. If the USB 3382 is configured as an Adapter (STRAP_RC_MODE=L), the PCI Express interface (Port 0) is enabled for enumeration by the PCI Express system. Otherwise, if the USB 3382 is configured as a Root Complex (STRAP_RC_MODE=H), the SuperSpeed USB is enabled for enumeration on the USB Host.
4. The 8051/USB Host or PCI Express Host initializes other USB and PCI Configuration registers.

4.2 PEX_PERST# Input

When asserted Low, the PEX_PERST# input resets all USB 3382 logic to its default state. This reset is referred to as *Fundamental Reset*. (Refer to Section 4.8 for further details.)

PEX_PERST# is typically connected to a power-on reset circuit. At power-on time, PEX_PERST# should remain asserted, for a minimum of 100 ms after all power supplies and clocks have stabilized. When PEX_PERST# is de-asserted, the USB 3382 reads the configuration information on its strapping inputs, and configures itself accordingly. If a serial EEPROM is present, the serial EEPROM contents are loaded into internal registers and/or 8051 Program memory. When Adapter mode is selected (STRAP_RC_MODE=L), the PCI Express interface is enabled for enumeration. The USB interface is not enabled until software enables the USB Port from the PCI Express interface. When Root Complex mode is selected (STRAP_RC_MODE=H), the USB interface is enabled for enumeration, and connected PCI Express Adapter (endpoint) devices attempt to linkup after PEX_PERST# input is de-asserted and after the serial EEPROM load is complete.

Note: *Throughout this data book, "Adapter mode" refers to both Legacy and Enhanced Adapter modes, unless specified otherwise.*

4.3 PWRON_RST# Input

The USB 3382 implements a Power-On Reset input, which is mainly used to reset the auxiliary power-operated circuit within the USB 3382. The USB 3382 has the option of entering a low-power-consumed Suspended state when the USB interface enters a Suspended state in Root Complex mode, or when the PCI Express Root Complex system enters the D3cold state in Adapter mode. In this PM state (Device Suspend state), most of the logic portion of the USB 3382 is powered down, except the circuitry that detects the USB Host wakeup from Suspend or Remote Wakeup from PCI Express Adapter (endpoint) devices. This reset must be held High when in the USB suspend state, or when the connected PCI Express system is in the D3cold state, depending upon the application. In such cases, PWRON_RST# input is derived from the Vaux supply on the PCI Express connector for Adapter mode operation, or from the USB_VBUS input in USB interface-powered Adapter and Root Complex mode applications. In systems that do not need to support this low-power state, PWRON_RST# input can be tied to the PEX_PERST# input. In such systems, the maximum allowed skew between the PEX_PERST# and PWRON_RST# inputs to the USB 3382 is 50 ns.

4.4 USB Host Reset

This section describes the causes of a USB Host Reset within the USB 3382.

4.4.1 *USB r2.0* Operating Mode

If the USB 3382 detects a single-ended zeros (SE0s) condition on the USB interface for more than 2.5 µs, the SE0 is interpreted as a Root Port Reset (USB Host Reset). This type of reset is recognized only when the following conditions exist:

- USB_VBUS input is High, and
- **USBCTL** register *USB Detect Enable* bit (USB Controller, offset 8Ch[3]) is Set

The following resources are reset:

- **OURADDR** register (USB Controller, offset A4h) and configuration
- Device Remote Wakeup enable and status

Root Port Reset does not affect the remainder of the Configuration registers. The **IRQSTAT1** register *Root Port Reset Interrupt Status* bit (USB Controller, offset 2Ch[4]) is Set when a Root Port Reset is detected. The CPU (8051 or PCI Express Host) takes appropriate action when this interrupt occurs.

According to the *USB r2.0*, the USB reset width is minimally 10 ms and can be longer, depending upon the upstream Host or hub. There is no specified maximum USB reset width.

When a *USB r2.0* Host Reset is received, Data packets that are sitting in the OUT FIFO are not flushed. However, packets that are sitting in the IN FIFO are flushed, and the IN FIFO pointers are reset to their defaults.

4.4.2 *USB r3.0* Operating Mode

There are two sources of USB Host Reset:

- Hot Reset received from the USB Host, with the *Reset* bit Set in the TS2 Ordered-Sets
- Warm Reset detected when Low-Frequency Periodic Signaling (LFPS) is received that meets the reset timing requirements, when not in the *SS.Disabled* substate

USB r2.0 and *USB r3.0* Hot Resets are treated the same in the USB 3382. Both cause a USB interface Reset when operating in their respective speed modes. When a USB Interface Reset is received, a device-specific **IRQSTAT1** register *Root Port Reset Interrupt Status* bit (USB Controller, offset 2Ch[4]) is Set, and the USB 3382 generates an interrupt to the 8051 or PCI Express Root Complex.

The following resources are reset:

- **OURADDR** register (USB Controller, offset A4h) and configuration
- Device remote wakeup enable and status
- U1, U2 enables
- U2 Inactivity timeout
- Function suspend
- Latency Tolerance Message (LTM) enables
- Isochronous delay (for further details, refer to the *USB r3.0*, Section 9.4.11)
- U1SEL, U2SEL, U1PEL, and U2PEL values (for further details, refer to the *USB r3.0*, Section 9.4.12)
- USB Link Error Counters
- Port Configuration (Enhanced Adapter and Root Complex modes)

When a *USB r3.0* Host Warm/Hot Reset is received, the USB 3382 *USB r3.0* Link Training and Status State Machine (LTSSM) enters the U0 state after the reset handshake successfully finishes. Additionally, when a Warm Reset is received, the USB 3382 *USB r3.0* LTSSM enters the *Rx Detect* state.

When a *USB r3.0* Hot/Warm Reset is received, Data packets that as sitting in the OUT FIFO are not flushed. However, packets that are sitting in the IN FIFO are flushed, and the IN FIFO pointers are reset to their defaults.

In addition to the above USB Host-initiated resets, the following all cause the same effect as the *USB r3.0* Warm Reset:

- Cable detach (USB cable's VBUS pin is disconnected, sensed by the USB_VBUS input)
- *USB r3.0 SS.Inactive* substate entry
- *USB r3.0* LTSSM U0-to-Disabled state transition
- **USBCTL** register *USB Detect Enable* bit (USB Controller, offset 8Ch[3]) is Cleared

4.5 PCI Express Reset

4.5.1 PCI Express Hot Reset

Note: *Applicable only in Adapter mode.*

PCI Express Hot Reset is an in-band Reset received on the upstream PCI Express interface, that uses *Control* bits that are Set in the TS2 training sequences. All *PCI Express Base r2.1*-related registers implemented in the USB 3382 are reset on Ports 0, 1, and 2. The USB OUT FIFO contents are flushed, and all pointers are reset to their default values. In Enhanced Adapter mode, the USB 3382 also propagates Hot Reset to the lone downstream PCI Express device through the PCI Express Link, using the Physical Layer (PHY) mechanism. The serial EEPROM is loaded, by default, when this reset occurs.

4.5.2 PCI Express Upstream Link Down

Note: *Applicable only in Adapter mode.*

PCI Express Upstream Link Down and Link Disabled events received on Port 0 cause the USB 3382 to reset. This has the same effect as a PCI Express Hot Reset. The serial EEPROM is loaded, by default, when this reset occurs.

4.5.3 Bridge Control Register Reset

Note: *Applicable only in Enhanced Adapter and Root Complex modes.*
Not applicable in Legacy Adapter mode, because there are no bridges in that mode.

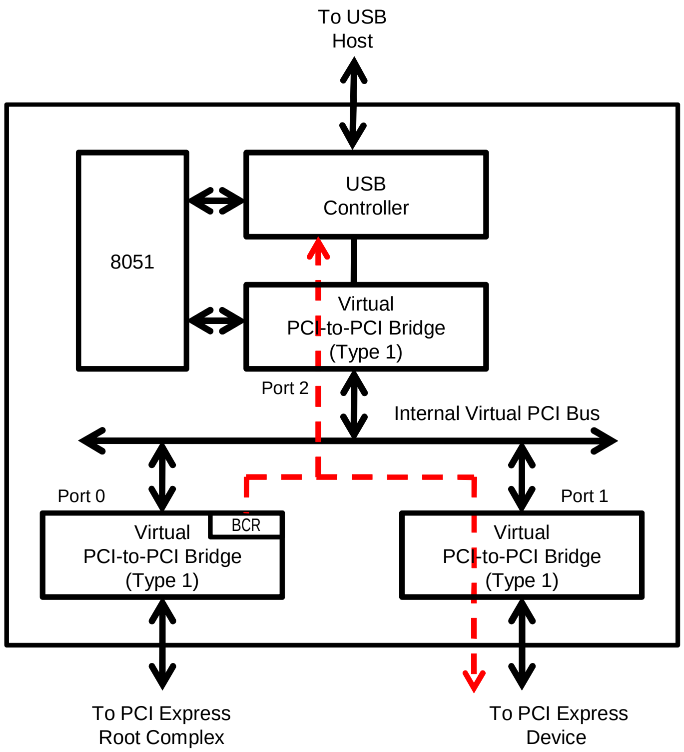
The Bridge Control Register (BCR) Reset is issued by software, by writing 1 to the **Bridge Control** register *Secondary Bus Reset* bit (All Ports, offset 3Ch[22]). This reset resets the bridges and endpoints below where the *Secondary Bus Reset* bit is Set. When the *Secondary Bus Reset* bit of a virtual bridge is Set, all bridges and endpoint devices on the virtual bridge's secondary side are reset. Only PCI Express logic and related registers are affected. In the USB 3382, this works differently for Enhanced Adapter and Root Complex modes, as described in the sections that follow.

4.5.3.1 Enhanced Adapter Mode

Reset Caused by Setting the Port 0 Bridge Control Reset Register

In the USB 3382, in Enhanced Adapter mode, reset can be performed by Setting Port 0's **Bridge Control** register *Secondary Bus Reset* bit (Port 0, offset 3Ch[22]). This causes the Port 1 and Port 2 PCI-to-PCI registers, as well as the USB Controller registers, to be reset. In Figure 4-1, the blocks intersected by the dashed red line are reset. Port 0 is not initialized by Setting this bit; however, its queues (to/from the affected Port(s)) are drained. The USB OUT FIFO contents are not flushed at this time. All RW and RW1C attribute registers in Port 1, Port 2, and the USB Controller are Cleared. Hot Reset is propagated to the lone downstream Port, Port 1, causing it to enter the DL_Down state. The USB OUT FIFO contents are flushed and pointers are reset, after the BCR is Cleared.

Figure 4-1. Enhanced Adapter Mode Bridge Control Reset, by Setting Port 0's Bridge Control Register *Secondary Bus Reset* Bit

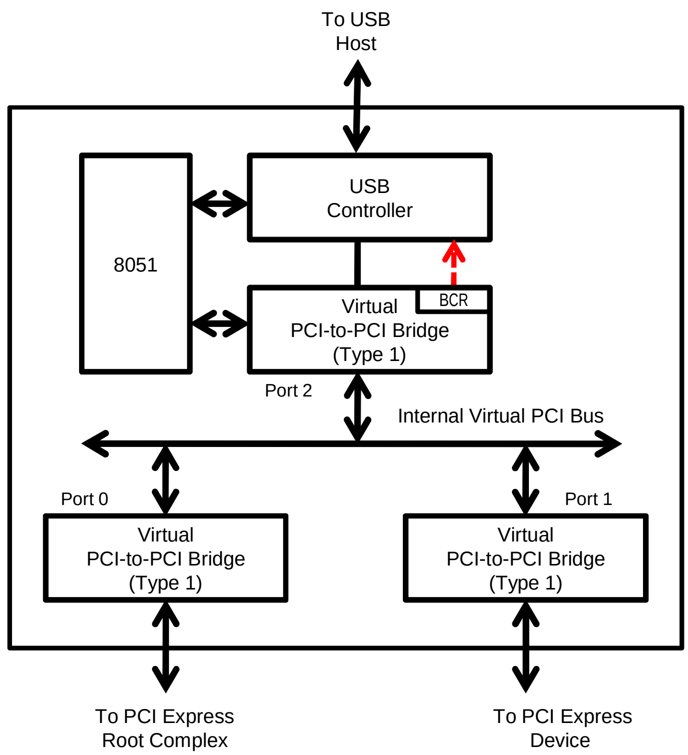


Reset Caused by Setting Port 2 Bridge Control Reset Register

In the USB 3382, in Enhanced Adapter mode, reset can also be performed by Setting Port 2's **Bridge Control** register *Secondary Bus Reset* bit (Port 2, offset 3Ch[22]). This causes all USB Controller registers that have RW and RW1C attributes to be reset. In Figure 4-2, the blocks intersected by the dashed red line are reset. The PCI-to-PCI bridges are not initialized by this reset; however, their queues [to/from the affected Port(s)] are drained. The USB OUT FIFO contents are flushed and pointers reset, after the *Secondary Bus Reset* bit is Cleared.

Another scenario here is Setting the *Secondary Bus Reset* bit on the Port 1 virtual bridge. This causes a Hot Reset to be propagated to the corresponding bridge endpoints, and the Link eventually goes into the *DL_Down* state. This does not reset any registers within the USB 3382. However, packets meant to go out on that Port in the *DL_Down* state are drained.

Figure 4-2. Reset Caused by Setting Port 2's Bridge Control Register *Secondary Bus Reset* Bit



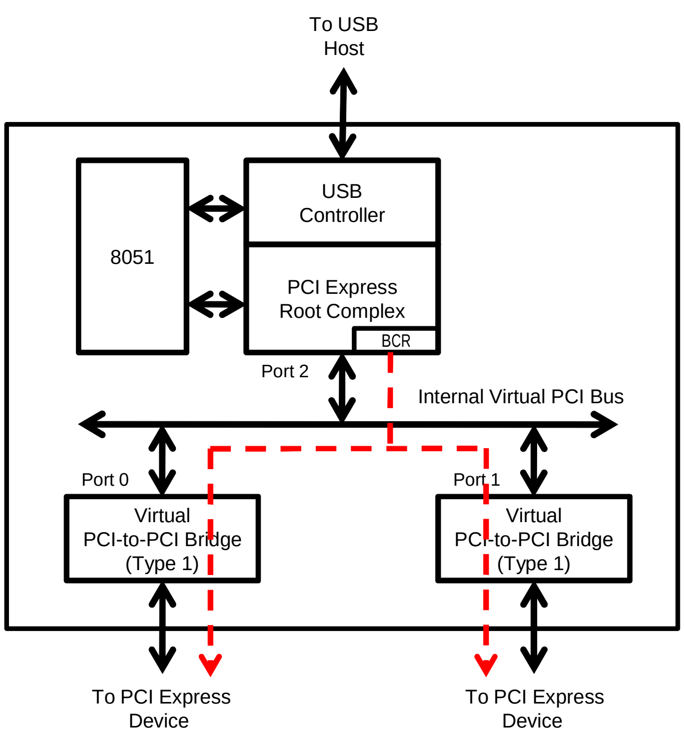
4.5.3.2 Root Complex Mode

Note: *Port 2 is an internal virtual PCI-to-PCI bridge that connects the USB Controller to the PCI Express fabric.*

In Root Complex mode (STRAP_RC_MODE=H), BCR Reset can occur when the USB Host/8051 Sets Port 2's **Bridge Control** register *Secondary Bus Reset* bit (Port 2, offset 3Ch[22]). This causes the Port 0 and Port 1 virtual bridge logic, as well as all Port 0 and Port 1 PCI Express-related registers with RW attributes, to be reset. Each of the secondary PCI-to-PCI bridges (downstream Ports) propagate an in-band Hot Reset onto its downstream Links. In addition, registers belonging to all secondary PCI-to-PCI bridges (downstream Ports) are initialized to their default values. In Figure 4-3, the blocks intersected by the dashed red line are reset.

The downstream Ports are held in the Reset state, until software removes the condition by Clearing Port 2's *Secondary Bus Reset* bit. Transaction Layer (TL) draining of non-empty queues to/from the affected Port(s) is handled similar to the case of that Port going to the *DL_Down* state, as defined in *PCI Express Base r2.1*, Section 2.9. After the BCR Reset is released, the PCI Express PHY goes into the *Detect* state, and starts the training sequence. The serial EEPROM is not re-loaded. Another scenario here is Setting Port 0 and Port 1's **Bridge Control** register *Secondary Bus Reset* bit (Port 0 and Port 1, offset 3Ch[22]). This causes a Hot Reset to be propagated to the corresponding bridge endpoints, and the Link eventually goes into the *DL_Down* state. This does not reset any USB 3382 registers; however, packets meant to go out on those Ports, in the *DL_Down* state, are drained.

Figure 4-3. Root Complex Mode Port 2 Bridge Control Register *Secondary Bus Reset* Propagation



4.6 Various Sources of Device-Specific Resets

Other than the reset sources explained in the previous sections, the USB 3382 also implements Configuration registers that can be Set by PCI Express Host software or the USB Host, to generate resets to various device functions. These reset sources, called *Soft Resets*, are software-generated by way of **DEVINIT** register bits (USB Controller, offset 00h[4:0]) register bits, and are explained, in detail, in the sections that follow.

4.6.1 8051 Soft Reset

An 8051 Soft Reset is performed by Setting the **DEVINIT** register *8051 Reset* bit (USB Controller, offset 00h[0]). This resets the 8051, and the 8051 is held in reset as long as the *8051 Reset* bit remains Set.

4.6.2 USB Soft Reset

A USB Soft Reset is performed by Setting the **DEVINIT** register *USB Soft Reset* bit (USB Controller, offset 00h[1]), which resets all USB-related logic and USB FIFOs within the USB 3382. This performs the same function as a USB Host Reset. (Refer to Section 4.4 for further details.)

4.6.3 PCI Express Soft Reset

A PCI Express Soft Reset is performed by Setting the **DEVINIT** register *PCI Soft Reset* bit (USB Controller, offset 00h[2]). This performs the same function as a PCI Express Hot Reset. (Refer to Section 4.5 for further details.)

4.6.4 Configuration Soft Reset

A Configuration Soft Reset is performed by Setting the **DEVINIT** register *Configuration Soft Reset* bit (USB Controller, offset 00h[3]), which Clears all Configuration registers within the USB 3382 and resets the registers to their default Fundamental Reset value.

4.6.5 FIFO Soft Reset

A FIFO Soft Reset is performed by Setting the **DEVINIT** register *FIFO Soft Reset* bit (USB Controller, offset 00h[4]), which resets the endpoint's FIFO controls and pointers, and flushes the endpoint's FIFO contents.

4.6.6 Reset Table

Table 4-1 summarizes the USB 3382 USB mode reset functionality.

Table 4-1. USB Mode Reset Table

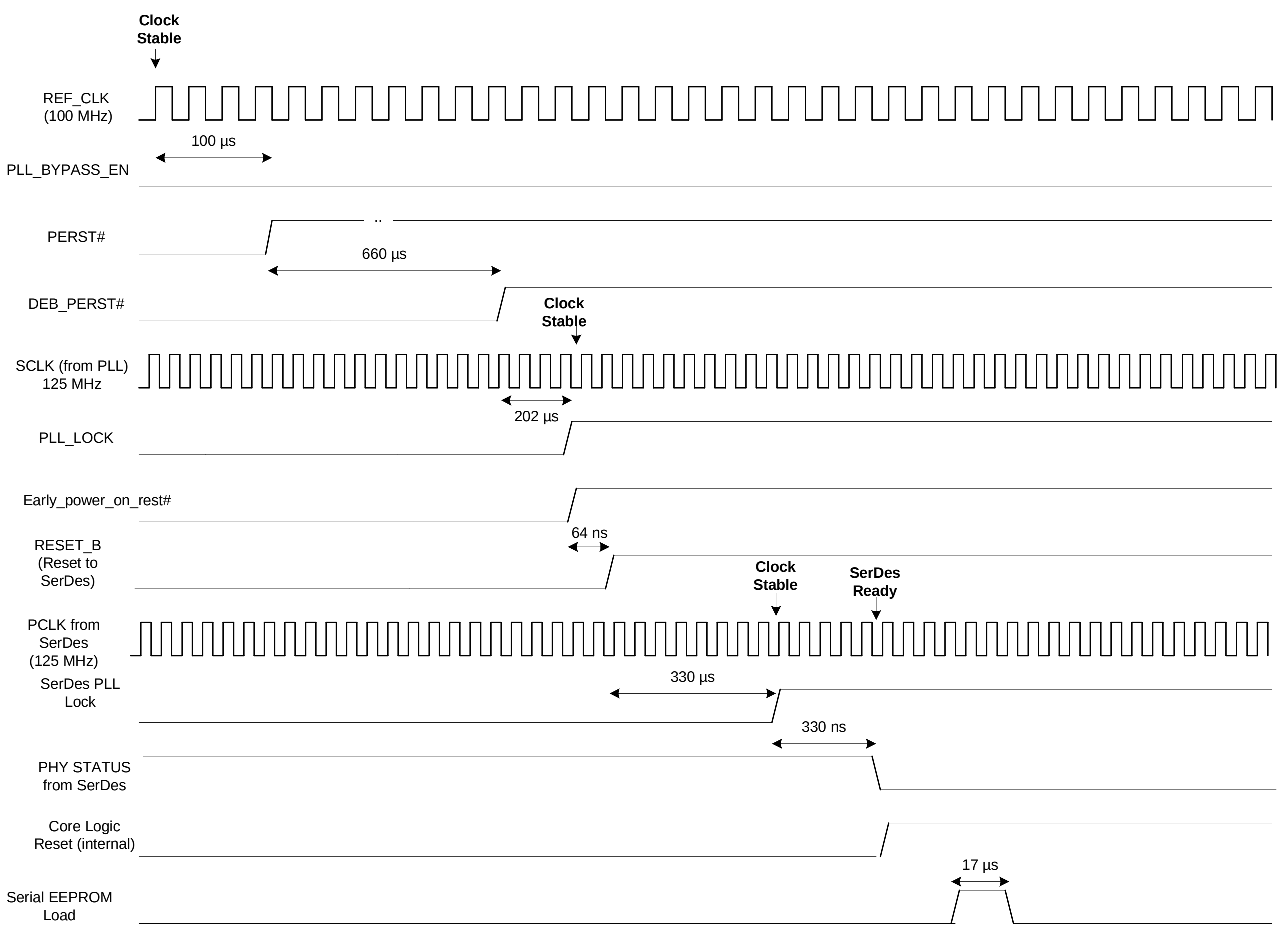
Reset Source	Device Resources								
	8051	USB PHY, LTSSM, and USB Port Configuration	*USB r2.0* or *USB r3.0* Core and Flags/CSRs Explained in Section 4.4	PCI Express Modules and PCI Express CSRs	Serial EEPROM Load	USB Interface CSRs	USB EP IN FIFOs	USB EP OUT FIFOs	Aux Power-Operated Logic
PEX_PERST# Input	✔	✔	✔	✔	✔	✔	✔	✔	
PWRON_RST# Input									✔
USB Root Port Reset (*USB r2.0)* –or– Hot Reset (*USB r3.0)*			✔				✔		
Warm Reset (*USB r3.0)*		✔	✔				✔		
PCI Express Hot Reset/ Link disable/Upstream *DL_Down*/Link L2/L3 Ready Link PM state to detect transition (Adapter mode only)				✔	✔			✔	
PCI Express Bridge Control Reset (Adapter mode)						✔ (USB Controller (Type 0) CSRs)		✔	
8051 Soft Reset	✔								
USB Soft Reset			✔				✔		
PCI Express Soft Reset				✔				✔	
Configuration Soft Reset						✔			
FIFO Soft Reset							✔	✔	

4.7 Reset and Clock Initialization Timing

Table 4-2. Reset and Clock Initialization Timing

Symbol	Description	Typical Delay
td1	REFCLK stable to PEX_Reset release time	100 µs
td2	PEX_Reset release to Reset de-bounce	660 µs
td3	Reset de-bounce to Phase-Locked Loop (PLL) Lock	202 µs
td4	Reset de-bounce to Core Reset release	331 µs
td5	Serial EEPROM load time with no serial EEPROM present	17 µs

Figure 4-4. Reset and Clock Initialization Timing (with Internal PLL)



4.8 Initialization Procedure

Upon exit from a Fundamental Reset, the USB 3382 initialization process executes, in the following sequence:

1. The USB 3382 samples the Strapping pin states, and configures hardware and register defaults accordingly.
2. If a serial EEPROM is present, serial EEPROM data is downloaded to the USB 3382 Configuration registers. (Refer to Section 4.9.)

   Alternatively, I^2C can be used to program all the registers (the same as would be done with the serial EEPROM), except, if PHY or DLL register values that affect SerDes parameters or Link initialization need to be changed, those registers must be programmed by serial EEPROM, so that the values are loaded prior to initial Link training. Because I^2C is relatively slow, the Links are usually up by the time the first I^2C Write occurs. The first I^2C command might be to block system access while the configuration is being changed, by disabling Port 0; the Ports can be disabled, by Setting the Port/SuperSpeed USB's **Port and SuperSpeed USB Control** register *Disable Port x/ SuperSpeed USB* bit (Port 0, offset 234h[19:16]).

   Configuration, including Port Configuration (Port 0, offset 574h[0]), can be changed by I^2C and/or serial EEPROM re-load. Changes take effect upon subsequent Hot Reset.
3. After configuration from the Strapping inputs, serial EEPROM, and/or I^2C is complete, the Physical Layer (PHY) of the configured PCI Express Ports attempts to bring up the PCI Express Links. After both components on a Link enter the initial Link Training state, the components proceed through PHY Link initialization and then through Flow Control initialization for VC0, preparing the DLL and TL to use the Link. Following Flow Control initialization for VC0, it is possible for VC0 TLPs and DLL Packets (DLLPs) to be transmitted across the Link.
4. Root Complex mode only – Additionally, after configuration is complete, the USB/SuperSpeed USB also attempts to train at this point.

4.9 Serial EEPROM Load upon Reset

By default, the USB 3382 starts loading serial EEPROM data after PEX_PERST# input is de-asserted. In Adapter mode, the USB 3382 also loads the serial EEPROM after the PCI Express Hot Reset sequence is performed or the Port 0 Link goes into the *DL_Down* state.

In Adapter mode, the USB 3382 also provides an additional *Control* bit to load the serial EEPROM upon PCI Express reset conditions. In this mode, other than Fundamental Reset, serial EEPROM load can be disabled or enabled by the **Debug Control** register *Disable Serial EEPROM Load on Hot Reset* bit (Port 0, offset 1DCh[17]).

When coming out of a Fundamental Reset, the 8051 is held in reset. After the serial EEPROM load starts and the serial EEPROM module detects a Program memory for the 8051, the reset module releases reset to the 8051 after the Program memory load is complete.

4.9.1 Initialization Summary when Using Serial EEPROM

Table 4-3 describes the USB 3382 initialization sequence, when using the optional serial EEPROM.

Table 4-3. Initialization Sequences

Status	Sequence
No Serial EEPROM, –or– Blank Serial EEPROM, –or– Invalid Serial EEPROM	• If a blank/no serial EEPROM is detected –or– a serial EEPROM with an invalid signature (valid signature is the first byte, value of 5Ah) is detected, serial EEPROM loading is ***not*** enabled. • In Adapter mode, do not automatically enable the USB interface – wait for the PCI Express Host to enumerate the PCI Express interface. The PCI Express Host enumerates the USB interface, configures the PCI Express interface (Port CSRs), and enumerates downstream PCI Express devices, by Setting the **USBCTL** register *USB Detect Enable* bit (USB Controller, offset 8Ch[3]). • When operating in Root Complex mode, enable the USB interface, using default register values. Wait for the USB Host to enumerate the USB interface, configure the PCI Express interface (Port CSRs), and enumerate downstream PCI Express devices. • Hold the 8051 in reset.
Valid Serial EEPROM with Configuration Register Data Only	• When operating in Root Complex mode, wait for the USB Host to program the PCI Express registers and enumerate PCI Express Adapter (endpoint) devices. • Enable the USB interface, using the register values loaded from the serial EEPROM (**USBCTL** register *USB Detect Enable* bit (USB Controller, offset 8Ch[3])). Optionally, when operating in Adapter mode, wait for the PCI Express Host to configure the USB Configuration registers. • Hold the 8051 in reset.
Valid Serial EEPROM with 8051 Program Memory Only	Start the 8051 and allow it to configure and enable the PCI Express device enumeration and USB interface register programming.
Valid Serial EEPROM with Configuration Register Data and 8051 Program Memory	• Enable the USB interface, using the register values loaded from the serial EEPROM, or wait for the 8051 to configure the USB registers and handle the enumeration. • Start the 8051 and allow it to configure the PCI Express and/or USB interfaces.

4.10 Default Port and SuperSpeed USB Configuration

The default Port Lane-width configuration is determined by the STRAP_PORTCFG input, which must be pulled High to VDD_IO or Low to Ground (refer to Table 4-4, which also includes the SuperSpeed USB Link width), to define the default device configuration. The configuration defined by the STRAP_PORTCFG input can be changed by downloading serial EEPROM data, or by I^2C programming (**Port Configuration** register *Port Configuration* bit (Port 0, offset 574h[0]) value), followed by a Hot Reset.

Table 4-4. Port/SuperSpeed Configurations

STRAP_PORTCFG Value	Port 0	Port 1	Port 2[a]	SuperSpeed USB
(default) 0	x1 Lane 0	x1 Lane 1	x1 Virtual Link	x1
1	x2 Lanes [0-1]		x1 Virtual Link	x1

a. Port 2 is an internal virtual PCI-to-PCI bridge that connects the USB Controller to the PCI Express fabric.

4.11 Default Register Initialization

Each USB 3382 Port/SuperSpeed USB defined in the Port/SuperSpeed USB Configuration process has its own set of assigned registers that control Port/SuperSpeed USB activities and status during standard operation. These registers are programmed to default/initial values, as detailed in:

- Chapter 15, "PCI Configuration Registers"
- Chapter 16, "USB Configuration Registers"

Following a Fundamental Reset, the basic PCI Express Support registers are initially programmed to the values specified in the *PCI Express Base r2.1*. The Device-Specific registers are programmed to the values specified in their register description tables. These registers can be changed by loading new data with the attached serial EEPROM, the I^2C Slave interface, and/or by CSR accesses using Configuration or Memory Writes; however, registers identified as RO ***cannot*** be modified by Configuration nor Memory Write Requests.

In Adapter mode, a valid serial EEPROM/8051, I^2C, and/or the PCI Express Host can be used to program the registers.

In Root Complex mode, serial EEPROM, I^2C, and/or the 8051 can also be used to program the registers. Alternatively after the USB Link is operational and the USB 3382 is configured, the USB Host can re-program the registers.

The USB 3382 supports the following mechanisms for accessing registers by way of the TL, as described in:

- Section 15.4.1, "PCI r3.0-Compatible Configuration Mechanism"
- Section 15.4.2, "PCI Express Enhanced Configuration Access Mechanism"
- Section 15.4.3, "Device-Specific Memory-Mapped Configuration Mechanism"

4.12 Device-Specific Registers

The PCI Configuration Device-Specific registers, detailed in the sections listed below, are unique to the USB 3382, and are not referenced in the *PCI Express Base r2.1*:

- Section 15.14, "Device-Specific Registers (Offsets 1C0h – 444h)"
- Section 15.15, "Device-Specific Registers (Offsets 530h – B88h)"

4.13 Serial EEPROM Load Time

Serial EEPROM initialization loads only the Configuration register data that is specifically programmed into the serial EEPROM. Registers that are not included in the serial EEPROM data are initialized to default register values.

Each register entry in the serial EEPROM consists of two Address bytes and four Data bytes (refer to Section 5.3, "Serial EEPROM Data Format"); therefore, each register entry (6 bytes, or 48 bits) requires 48 serial EEPROM clocks to download. Thus, at the serial EEPROM clock default frequency of 1 MHz, after initial overhead to read the **Serial EEPROM Status** register (Port 0, offset 260h) (16 serial EEPROM clocks, or 16 µs), plus another 40 serial EEPROM clocks (40 µs) to begin reading the register data, each register entry in the serial EEPROM requires 48 µs to download. A serial EEPROM containing 50 register entries (typical configuration, assuming the serial EEPROM is programmed only with non-default register values) and clocked at 1 MHz, takes approximately 2.5 ms to load (16 + 40 + (48 x 50 µs) = 2,456 µs).

To reduce the serial EEPROM initialization time, the first register entry in the serial EEPROM can increase the clock frequency, by programming the **Serial EEPROM Clock Frequency** register (Port 0, offset 268h) to a value of 2h (5 MHz), or 3h (9.62 MHz), if the serial EEPROM supports the higher frequency at the serial EEPROM supply voltage (typically 3.3V). At 5-MHz clocking, the serial EEPROM load time for 50 register entries can be reduced to approximately 575 µs. Because the *PCI Express Base r2.1* allows a 20-ms budget for system hardware initialization, the default 1-MHz serial EEPROM clock is often sufficient when the quantity of Ports and registers programmed by serial EEPROM is relatively small.

For further details, refer to Chapter 5, "Serial EEPROM Controller."

4.13.1 I^2C Load Time

Initialization using I^2C is slower than serial EEPROM initialization, because the I^2C Slave interface operates at a lower clock frequency (100 KHz maximum) and the quantity of bits per Register access is increased (because the Device address is included in the bit stream). Writing one register using 100-KHz clocking takes approximately 830 µs (83 clock periods).

For further details, refer to Section 12.2, "I^2C Slave Interface."

Chapter 5 Serial EEPROM Controller

5.1 Overview

The USB 3382 provides an interface to Serial Peripheral Interface (SPI)-compatible serial EEPROMs. The interface consists of a Chip Select, Clock and Write Data outputs, and a Read Data input, and operates at a programmable frequency of up to 17.86 MHz. Compatible 8 x 8 KB serial EEPROMs are the Atmel AT25640A, ON Semiconductor CAT25640, or equivalent. The USB 3382 supports up to a 16-Mbit serial EEPROM, that uses 1-, 2-, or 3-byte addressing (2-byte addressing is recommended); the USB 3382 automatically determines the appropriate addressing mode.

5.2 Features

- Detection of whether a serial EEPROM is present/not present
- Supports high-speed serial EEPROMs with Serial Peripheral Interface (SPI) interface
- Non-volatile storage for register default values loaded during Power-On Reset
- 4-byte Read/Write access to the serial EEPROM, through Port 0
- Serial EEPROM data format allows for loading registers by Port/Address location
- Required serial EEPROM size is dependent upon the number of registers being changed
- Automatic support for 1-, 2-, or 3-byte-addressable serial EEPROMs
- Manual override for quantity of serial EEPROM Address bytes
- Programmable serial EEPROM clock frequency
- Programmable serial EEPROM clock-to-chip select timings
- No Cyclic Redundancy Check (CRC), single *Valid* byte at the start of serial EEPROM memory

5.3 Serial EEPROM Data Format

The data in the serial EEPROM is stored in the format defined in Table 5-1. The serial EEPROM Format Byte at Serial EEPROM address 1h is organized as listed in Table 5-2.

The format of the REGADDR bytes, as defined in Table 5-1, is described in Table 5-3. This field selects which block of registers within the USB 3382 to be accessed, and the DWord offset to the specific register within that register block.

Table 5-1. Serial EEPROM Data Format

Location	Value	Description
0h	5Ah	Validation Signature
1h	Refer to Table 5-2	Serial EEPROM Format Byte
2h	REG_BYTE_COUNT (LSB)	Configuration register Byte Count (LSB)
3h	REG_BYTE_COUNT (MSB)	Configuration register Byte Count (MSB)
4h	REGADDR (LSB)	1st Configuration Register Address (LSB)
5h	REGADDR (MSB)	1st Configuration Register Address (MSB)
6h	REGDATA (Byte 0)	1st Configuration Register Data (Byte 0)
7h	REGDATA (Byte 1)	1st Configuration Register Data (Byte 1)
8h	REGDATA (Byte 2)	1st Configuration Register Data (Byte 2)
9h	REGDATA (Byte 3)	1st Configuration Register Data (Byte 3)
Ah	REGADDR (LSB)	2nd Configuration Register Address (LSB)
Bh	REGADDR (MSB)	2nd Configuration Register Address (MSB)
Ch	REGDATA (Byte 0)	2nd Configuration Register Data (Byte 0)
Dh	REGDATA (Byte 1)	2nd Configuration Register Data (Byte 1)
Eh	REGDATA (Byte 2)	2nd Configuration Register Data (Byte 2)
Fh	REGDATA (Byte 3)	2nd Configuration Register Data (Byte 3)
…	…	…
REG BYTE COUNT + 4	MEM COUNT (LSB)	8051 Program Memory Byte Count (LSB)
REG BYTE COUNT + 5	MEM COUNT (MSB)	8051 Program Memory Byte Count (MSB)
REG BYTE COUNT + 6	PROG (Byte 0)	First Byte of 8051 Program Memory
REG BYTE COUNT + 7	PROG (Byte 1)	Second Byte of 8051 Program Memory
…	…	…
FFFFh	PROG (Byte *n*)	Last Byte of 8051 Program Memory

Table 5-2. Serial EEPROM Format Byte

Bit(s)	Description
0	**Configuration Register Load** 0 = Serial EEPROM locations 2h and 3h are read, to determine the quantity of 32-bit Configuration registers to be loaded 1 = Configuration register load is disabled
1	**Program Load** 1 = 8051 program memory is loaded from the serial EEPROM, starting at location REG BYTE COUNT + 6. The quantity of bytes to load is determined by the value in serial EEPROM locations REG BYTE COUNT + 4 and REG BYTE COUNT + 5.
2	**8051 Enable** 1 = 8051 Reset is de-asserted after its Program memory is loaded from the serial EEPROM. Valid only when bit 1 (*Program Load*) is Set.
7:3	***Reserved***

Table 5-3. REGADDR[15:0] Address Format

Bit(s)	Description
9:0	**Register Address Bits [11:2]** Register Address field specifies the DWord offset to the specific register of the selected register block.
12:10	**Port Selector Bits [2:0]** Selects the register block to be accessed. 000b = Port 0 (Enhanced Adapter mode) / USB Controller PCI Configuration registers (Type 0) (Legacy Adapter mode) 001b = Port 1 (Enhanced Adapter mode) 010b = Port 2 (Enhanced Adapter mode) 011b = USB Controller PCI Configuration registers (Type 0) (Enhanced Adapter mode) 100b = USB Controller Configuration registers All other encodings are ***reserved***.
15:13	***Reserved*** Should be Cleared.

5.4 Serial EEPROM Initialization

After PEX_PERST# is de-asserted, the USB 3382 attempts to detect and load content from an external serial EEPROM. A pull-down resistor on the EE_RDDATA/EE_DO output produces a value of 00h if a serial EEPROM is not installed. If a serial EEPROM is detected, the first byte (validation signature) is read. If a value of 5Ah is read, it is assumed that the EEPROM is programmed for the USB 3382. The serial EEPROM address width is determined while this first byte is being read. If the first byte is not 5Ah, the serial EEPROM is blank or programmed with invalid data. In this case, the 8051 is held in reset, and either the USB or PCI Express interface is enabled for enumeration, depending upon whether STRAP_RC_MODE is pulled High to VDD_IO or pulled or tied Low to Ground, respectively. Also, the **Serial EEPROM Status** register *EepAddrWidth* field (Port 0, offset 260h[23:22]) reports a value of 00b (undetermined width).

If the serial EEPROM contains valid data, the second byte (Serial EEPROM Format byte) is read to determine which serial EEPROM sections should be loaded into the USB 3382 Configuration registers and memory. If bit 0 is Set, Bytes 2 and 3 determine the quantity of serial EEPROM locations that contain Configuration register addresses and data. Each Configuration register entry consists of two bytes of register address (bit 10 Low selects the PCI Configuration registers, and bit 10 High selects the Memory-Mapped Configuration registers) and four bytes of register Write data. If bit 1 of the Serial EEPROM Format byte is Set, locations REG BYTE COUNT + 4 and REG BYTE COUNT + 5 are read, to determine the byte quantity to transfer from the serial EEPROM into the 8051 Program RAM. After this transfer completes, and if bit 2 in the Serial EEPROM Format byte is Set, the 8051 reset is de-asserted, allowing the 8051 to execute the firmware.

The EE_CLK/EE_SK output clock frequency is determined by the **Serial EEPROM Clock Frequency** register *EepFreq[2:0]* field (Port 0, offset 268h[2:0]). The default clock frequency is 1 MHz. At this clock rate, it takes approximately 48 µs per DWORD during Configuration register initialization. For faster loading of large serial EEPROMs that support a faster clock, the first Configuration register load from the serial EEPROM could be to the **Serial EEPROM Clock Frequency** register.

5.5 PCI Express Configuration, Control, and Status Registers

The PCI Express Configuration, Control, and Status registers that can be initialized are detailed in Chapter 15, “PCI Configuration Registers.”

5.6 Serial EEPROM Registers

The Serial EEPROM register (Port 0, offsets 260h through 26Ch) parameters defined in Section 15.14.3, “Device-Specific Registers – Serial EEPROM (Offsets 260h – 26Ch),” can be changed, using the serial EEPROM. At the last serial EEPROM entry, the **Serial EEPROM Status and Control** register (Port 0, offset 260h) can be programmed to issue a Write Status (WRSR) command, to enable the write protection feature(s) within the serial EEPROM data, if needed.

5.7 Serial EEPROM Random Read/Write Access

A USB Host, the 8051, or a Master device on the PCI Express interface can use the **Serial EEPROM Control** register (Port 0, offset 260h) to access the serial EEPROM. This register contains 8-bit Read and Write Data fields, Read and Write Start signals, and related Status bits.

The following "C" routines demonstrate the firmware protocol required to access the serial EEPROM by way of the **Serial EEPROM Control** register. An interrupt is generated when the register's *Serial EEPROM Busy* bit changes from True to False.

Note: *In the "C" routines that follow, the* ***Serial EEPROM Control*** *register is referred to as "EECTL".*

5.7.1 Serial EEPROM Opcodes

```
READ_STATUS_EE_OPCODE = 5
WREN_EE_OPCODE = 6
WRITE_EE_OPCODE = 2
READ_EE_OPCODE = 3
```

5.7.2 Serial EEPROM Low-Level Access Routines

```
int EE_WaitIdle()
{
   int eeCtl, ii;
   for (ii = 0; ii < 100; ii++)
   {
        USB 3382Read(EECTL, eeCtl);              /* read current value in EECTL */
        if ((eeCtl & (1 << EEPROM_BUSY)) == 0)  /* loop until idle */
        return(eeCtl);
    }
    PANIC("EEPROM Busy timeout!\n");
}

void EE_Off()
{
    EE_WaitIdle();                               /* ensure EEPROM is idle */
    USB 3382Write(EECTL, 0);                     /* turn off everything
                                                 (especially EEPROM_CS_ENABLE) */
}

int EE_ReadByte()
{
    int eeCtl = EE_WaitIdle();                   /* ensure EEPROM is idle */
    eeCtl |= (1 << EEPROM_CS_ENABLE) |
             (1 << EEPROM_BYTE_READ_START);
    USB 3382Write(EECTL, eeCtl);                 /* start reading */
    eeCtl = EE_WaitIdle();                       /* wait until read is done */
    return((eeCtl >> EEPROM_READ_DATA) & FFh); /* extract read data from EECTL */
}

void EE_WriteByte(int val)
{
    int eeCtl = EE_WaitIdle();                   /* ensure EEPROM is idle */
    eeCtl &= ~(FFh << EEPROM_WRITE_DATA);       /* clear current WRITE value */
    eeCtl |= (1 << EEPROM_CS_ENABLE) |
             (1 << EEPROM_BYTE_WRITE_START) |
             ((val & FFh) << EEPROM_WRITE_DATA);
    USB 3382Write(EECTL, eeCtl);
}
```

5.7.3 Serial EEPROM Read Status Routine

```
…
EE_WriteByte(READ_STATUS_EE_OPCODE);            /* read status opcode */
status = EE_ReadByte();                         /* get EEPROM status */
EE_Off();                                       /* turn off EEPROM */
…
```

5.7.4 Serial EEPROM Write Data Routine

```
…
EE_WriteByte(WREN_EE_OPCODE);                   /* must first write-enable */
EE_Off();                                       /* turn off EEPROM */
EE_WriteByte(WRITE_EE_OPCODE);                  /* opcode to write bytes */
#ifdef THREE_BYTE_ADDRESS_EEPROM                /* three-byte addressing EEPROM?
*/
   EE_WriteByte(addr >> 16);                    /* send high byte of address */
#endif
EE_WriteByte(addr >> 8);                        /* send next byte of address */
EE_WriteByte(addr);                             /* send low byte of address */
for (ii = 0; ii < n; ii++)
{
   EE_WriteByte(buffer[ii]);                    /* send data to be written */
}
EE_Off();                                       /* turn off EEPROM */
….
```

5.7.5 Serial EEPROM Read Data Routine

```
…
EE_WriteByte(READ_EE_OPCODE);                   /* opcode to write bytes */
#ifdef THREE_BYTE_ADDRESS_EEPROM                /* three-byte addressing EEPROM?
*/
   EE_WriteByte(addr >> 16);                    /* send high byte of address */
#endif
EE_WriteByte(addr >> 8);                        /* send next byte of address */
EE_WriteByte(addr);                             /* send low byte of address */
for (ii = 0; ii < n; ii++)
{
   buffer[ii] = EE_ReadByte(buffer[ii]);        /* store read data in buffer */
}
EE_Off();                                       /* turn off EEPROM */
```

Chapter 6 8051 Micro-Controller Unit

6.1 Overview

The embedded 8051 Micro-Controller Unit (MCU) allows the USB 3382 to operate as a stand-alone system-on-chip (SOC) processor for enumerating and servicing PCI Express devices, as well as handling USB endpoints, DMA, and interrupt events. The 8051 controls 32 KB of on-chip RAM that is shared between program and data. It also controls 256 bytes of internal RAM (IRAM) space for MCU registers, Stack, and Scratchpad space. A debug Port is used for 8051 firmware development. The MCU executes machine cycles in an average of 2.39 clock periods and operates at 125 MHz, thus providing over 50 MIPs.

6.2 8051 Memory Map

The 8051 has access to 32 KB of memory, to be shared between the Program and Data space.

6.2.1 Program and Data Space (64 KB)

The program memory consists of 32 KB of RAM, which can be written to by the serial EEPROM, USB, and/or PCI Express interface.

Table 6-1. Program and Data Space (64 KB)

Address	Device
0000h – 7FFFh	Program RAM (32 KB, shared with external Data space)
8000h – FFFFh	***Not used***

6.2.2 IRAM (256 Bytes)

IRAM is 8-bit wide memory that is directly accessible to 8051 instructions (CPU register space), and divided into two 128-byte regions – Low and High (refer to Table 6-2):

- **Low IRAM region (Addresses 0000h through 007Fh)** – Can be accessed using direct (MOV A, direct) or indirect (MOV A, @Ri) addressing. (Refer to Table 6-3.)
- **High IRAM region (Addresses 0080h through 00FFh)** – When accessed using direct addressing, the region accesses a block of Cursor registers (Special Function registers (SFRs); refer to Section 6.2.2.1), the 8051 can access the 32-KB Code/Data memory, Port, and USB Controller registers, USB endpoint FIFOs, and the PCI Express interface.

  When accessed using indirect addressing, the region accesses 128 bytes of general-purpose RAM. In this way, two 128-byte blocks occupy the same Address range.

Table 6-2. IRAM (256 Bytes)

Address	Device
0000h – 007Fh	Low RAM region (128 bytes, direct or indirect addressed)
0080h – 00FFh	High RAM region, Special Function registers (128 bytes, direct addressed)
	High RAM region, general-purpose RAM (128 bytes, indirect addressed)

Table 6-3. Low IRAM Region (128 Bytes, Direct or Indirect Addressing)

Address	Device
00h – 1Fh	CPU Register set
20h – 2Fh	16 Bytes General-Purpose RAM (Bit addressable)
30h – 7Fh	80 bytes general-purpose RAM (Stack and Scratchpad)

6.2.2.1 Special Function Registers

These Special Function registers (SFRs) are internal to the 8051, and are listed here, in Table 6-4, for reference only.

Table 6-4. Special Function Registers (High IRAM Region, Direct Addressed)

Address	Register	Description	USB 3382 Unique	Bit Addressable
80h	P0	Port 0		✔
81h	SP	Stack Pointer		
82h	DPL	Data Pointer Low Byte		
83h	DPH	Data Pointer High Byte		
84h	DPL1	Data Pointer 1 Low Byte		
85h	DPH1	Data Pointer 1 High Byte		
86h	DPS	Data Pointer Select		
87h	PCON	Power Control		
88h	TCON	Timer/Counter Control		✔
89h	TMOD	Timer/Counter Mode Control		
8Ah	TL0	Timer/Counter 0 Low Byte		
8Bh	TL1	Timer/Counter 1 Low Byte		
8Ch	TH0	Timer/Counter 0 High Byte		
8Dh	TH1	Timer/Counter 1 High Byte		
8Eh	CKCON	Clock Control		
90h	P1	Port 1		x
98h	SCON	Debug Port Control		✔
99h	SBUF	Debug Port Data Buffer		
A0h	P2	Port 2 (Virtual PCI-to-PCI bridge)		✔
A8h	IE	Interrupt Enable Control		✔
A9h	SADDR	Debug Port Target Address		
B0h	P3	USB Controller (PCI Express endpoint)		✔
B8h	IP	Interrupt Priority Control		✔
B9h	SADEN	Debug Port Automatic Address Recognition Enable		
C4h	PMR	Power Management		
C7h	TA	Timed Access		
C8h	T2CON	Timer/Counter 2 Control		✔
C9h	T2MOD	Timer/Counter 2 Mode Control		
CAh	RCAP2L	Timer/Counter 2 Capture (LSB)		
CBh	RCAP2H	Timer/Counter 2 Capture (MSB)		
CCh	TL2	Timer/Counter 2 Low Byte		
CDh	TH2	Timer/Counter 2 High Byte		

Table 6-4. Special Function Registers (High IRAM Region, Direct Addressed) *(Cont.)*

Address	Register	Description	USB 3382 Unique	Bit Addressable
D0h	PSW	Program Status Word		✔
E0h	ACC	Accumulator		✔
E8h	IRQ0A	IRQA Status (GPEP – IN; LSB)	✔	✔
E9h	IRQ0C	IRQC Status (GPEP – OUT; MSB)	✔	✔
F0h	B	B Register		✔
F2h	CFGCTL0	Configuration Register Control Byte 0	✔	
F3h	IFSTAT	Interface Status		
F4h	CFGADDR0	Configuration Register Address Byte 0 (LSB)	✔	
F5h	CFGADDR1	Configuration Register Address Byte 1 (MSB)	✔	
F8h	IRQ0B	IRQB Status	✔	✔
F9h	CFGDATA0	Configuration Register Data Byte 0 (LSB)	✔	
FAh	CFGDATA1	Configuration Register Data Byte 1	✔	
FBh	CFGDATA2	Configuration Register Data Byte 2	✔	
FCh	CFGDATA3	Configuration Register Data Byte 3 (MSB)	✔	

6.3 Device-Specific Special Function Registers

6.3.1 Configuration Register Access Special Function Registers

A set of eight Special Function registers (SFRs; refer to Table 6-5) allows the 8051 to access the USB 3382 Configuration registers, USB endpoints, and external PCI Express devices (by way of the PCI Master registers (USB Controller, offsets 100h through 10Ch, and 11Ch)). For a Configuration register access, the 8051 first sets up the register Address and Data (for a Write, at IRAM addresses F4h and F5h, and F9h through FCh, respectively). Then, the Control byte is written, with the *Start* bit Set. The remaining fields in the Control byte are formatted, as listed in Table 6-5.

The *Start* bit is automatically Cleared when the transaction is complete. The *Read/Write* bit determines the value returned in the **CFGDATA*x*** registers, as listed in Table 6-5.

Configuration or *PCI Express Access Busy* bits can also be monitored in the bit-addressable **IRQ0B** SFR (8051, Address F8h).

Table 6-5. Configuration Register Access Special Function Registers

IRAM Address	Register			
	Bit(s)	**Description**	**Attribute**	**Default**
F2h	**CFGCTL0** (Configuration Register Control Byte 0)			
	3:0	**Byte Enables** Selects which of the four bytes are written.	RW	0h
	5:4	**Space Select** 00b = PCI Configuration registers 01b = Memory-Mapped Configuration registers All other encodings are ***reserved***.	RW	00b
	6	**Start** Automatically Cleared.	RW	0
	7	**Read/Write** 0 = Write 1 = Read	RW	0
F3h	**IFSTAT** (Interface Status)			
	0	**Error Status** 1 = PCI Express or USB cycle started by the 8051 has received an error. If the error is a UR, the Data registers are programmed to FFFF_FFFFh. Write 1 to Clear.	RW	0
	1	**CPL Data Valid** 1 = Completion data is in the Data registers. F9h through FCh. Write 1 to Clear.	RW	0
	7:2	***Reserved***	R	0-0h
F4h	7:0	**CFGADDR0** (Configuration Register Address Byte 0 (LSB))		
F5h	7:0	**CFGADDR1** (Configuration Register Address Byte 1 (MSB))		
F9h	7:0	**CFGDATA0** (Configuration Register Data Byte 0 (LSB)) 0 = Write data 1 = Read data returned from Configuration Read transaction		
FAh	7:0	**CFGDATA1** (Configuration Register Data Byte 1) 0 = Write data 1 = Read data returned from Configuration Read transaction		
FBh	7:0	**CFGDATA2** (Configuration Register Data Byte 2) 0 = Write data 1 = Read data returned from Configuration Read transaction		
FCh	7:0	**CFGDATA3** (Configuration Register Data Byte 3 (MSB)) 0 = Write data 1 = Read data returned from Configuration Read transaction		

6.3.2 Interrupt Status Special Function Registers

A set of three, bit-addressable SFRs allows the 8051 to read the USB 3382 **IRQSTAT0** register (USB Controller, offset 28h) *Interrupt Status* bits. Also, the Busy status of a Configuration register or PCI Master access is monitored. Usually, these transactions are completed before the 8051 tests these bits.

Table 6-6. Interrupt Status Special Function Registers

IRAM Address	Register Name
E8h	**IRQ0A** (IRQA Status (GPEP – IN; LSB))
E9h	**IRQ0C** (IRQC Status (GPEP – OUT; MSB))
F8h	**IRQ0B** (IRQB Status)

6.3.3 PCI Express Accesses

The 8051 can initiate accesses to PCI Express space. PCI Express accesses must be indirectly performed through the USB 3382 **PCI Master Control** registers. The **PCIMSTADDR** register (USB Controller, offset 104h) determines the base PCI address of these accesses, and data is read or written through the **PCIMSTDATA** register (USB Controller, offset 108h). Bits in the **PCIMSTCTL** register (USB Controller, offset 100h) determine the transaction direction, start the transaction, and detect when the transaction is complete. The **PCIMSTADDR** register determines the PCI address when performing Type 0 or Type 1 Configuration cycles.

The **PCIMSTADDR**, **PCIMSTDATA**, and **PCIMSTCTL** registers are also used by the USB interface, to access the PCI Express interface. The PCIOUT and PCIIN Dedicated endpoints allow these registers to be accessed with one USB transaction. There is no Configuration register resource locking during PCI Master cycles; therefore, the USB Host and 8051 are required to negotiate PCI Express interface control.

For further details, refer to Section 7.4.1.1, "PCI Master Control Registers," and Section 16.7, "PCI Express/Configuration Cursor Registers."

6.4 8051 Interrupts

The 8051 can service interrupts from internal resources, or USB or PCI Express sources. Individual interrupts are enabled to the 8051, using the USB 3382 **CPUIRQENB0** and **CPUIRQENB1** registers (USB Controller, offsets 18h and 1Ch, respectively). Interrupts from the USB 3382 **IRQSTAT0** register (USB Controller, offset 28h) are routed to the 8051's Interrupt Input 0, while interrupts from the USB 3382 **IRQSTAT1** register (USB Controller, offset 2Ch) are routed to the 8051's Interrupt Input 1.

Chapter 7 PCI Express Interface

7.1 Overview

This chapter describes the USB 3382's PCI Express functions. With respect to PCI Express functions, the USB 3382 provides three modes of operation – Legacy Adapter, Enhanced Adapter, and Root Complex. The combined STRAP_LEGACY and STRAP_RC_MODE input states determine the mode of operation, as listed in Table 7-1. Each mode is described in the sections that follow.

In addition to the modes of operation, PCI Express Accesses to On-Chip RAM are also discussed.

Note: *Throughout this data book, "Adapter mode" refers to both Legacy and Enhanced Adapter modes, unless specified otherwise.*

Table 7-1. Modes of Operation and Related Strapping Input States

Mode	STRAP_LEGACY Input State	STRAP_RC_MODE Input State
Legacy Adapter Mode	H	L
Enhanced Adapter Mode	L	L
Root Complex Mode	L	H

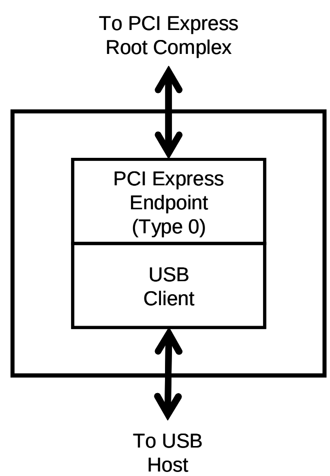
7.2 Legacy Adapter Mode

The USB 3382 is configured in Legacy Adapter mode when the following conditions exist:

- STRAP_LEGACY input is pulled High to VDD_IO, and
- STRAP_RC_MODE input is pulled or tied Low to Ground

In this mode, the USB 3382 appears as a PCI Express Adapter (endpoint), presenting a Type 0 PCI device Header to the system, as illustrated in Figure 7-1. When in Legacy Adapter mode, the USB 3382 presents a register set that is compatible with PLX's NETCHIP NET 2282 PCI to High-Speed USB 2.0 Controller. In this way, application programs that were originally written for use with the NET 2282 can be migrated to USB 3382 designs, with little or no modification.

Figure 7-1. Legacy Adapter Mode



7.3 Enhanced Adapter Mode

The USB 3382 is configured in Enhanced Adapter mode when the STRAP_LEGACY and STRAP_RC_MODE inputs are both pulled or tied Low to Ground. In this mode, the USB 3382 appears to the PCI Express Root Complex as a PCI Express switch with a USB Controller (PCI Express endpoint) attached to downstream Port 2 of the switch, as illustrated in Figure 7-2.

Note: *Port 2 is an internal virtual PCI-to-PCI bridge that connects the USB Controller to the PCI Express fabric.*

The USB 3382 supports two different Port configurations, configurable using the STRAP_PORTCFG input, or by serial EEPROM. When the STRAP_PORTCFG input is pulled or tied Low to Ground, the USB 3382 presents both upstream and downstream PCI Express Ports, both with x1 Links, as illustrated in Figure 7-2. When the STRAP_PORTCFG input is pulled High to VDD_IO, the USB 3382 presents a single upstream PCI Express Port x2 Link, as illustrated in Figure 7-3.

When in Enhanced Adapter mode, the USB Controller presents an extended set of registers and functions that support operation at SuperSpeed *USB r3.0* data rates. These functions are described in Chapter 8, "USB Controller Functional Description."

Figure 7-2. Enhanced Adapter Mode with Downstream PCI Express Port

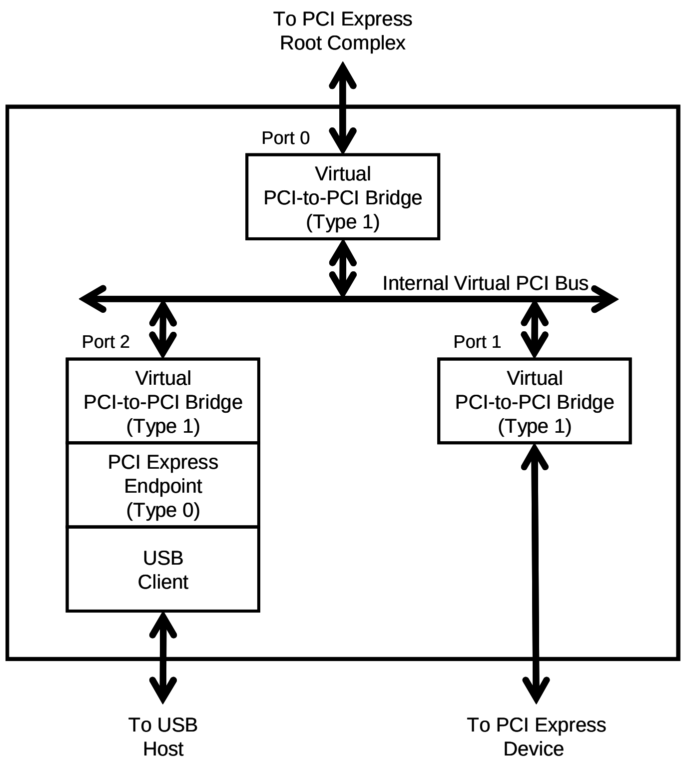
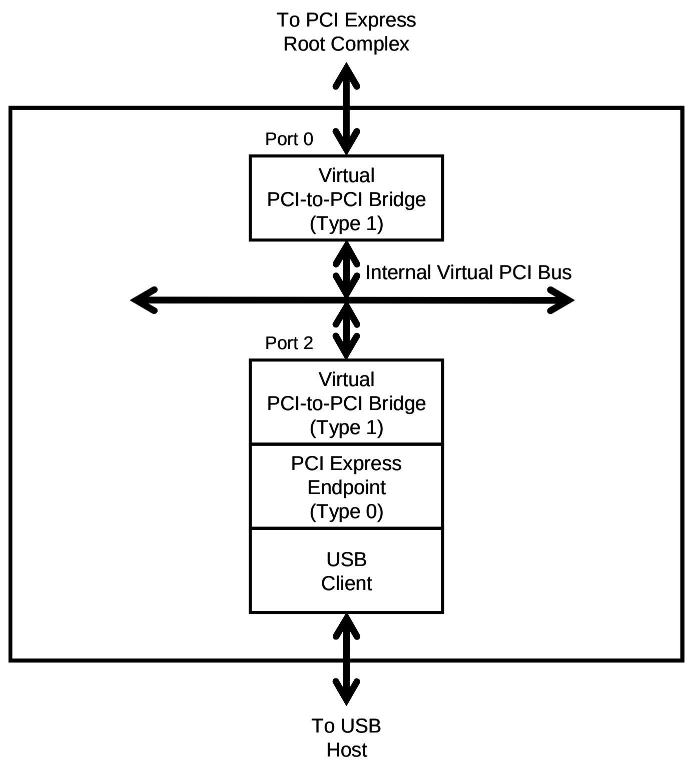


Figure 7-3. Enhanced Adapter Mode with Single PCI Express Port x2 Link



7.3.1 PCI Express Adapter (Endpoint)

The USB Controller presents a Type 0 Configuration Header to the system, providing Base Address registers (BARs) that allow the PCI Express Root Complex CPU to access internal registers, internal RAM (DMA Descriptor space), and USB endpoint FIFOs. In Enhanced Adapter mode, the USB Controller resides behind a series of virtual PCI-to-PCI bridges, as illustrated in Figure 7-2 and Figure 7-3.

7.4 Root Complex Mode

The USB 3382 is configured in Root Complex mode when the following conditions exist:

- STRAP_LEGACY input is pulled or tied Low to Ground, and
- STRAP_RC_MODE input is pulled High to VDD_IO

In this mode, the USB 3382 is configured as a PCI Express Root Complex, as illustrated in Figure 7-4. When the USB 3382 is configured as a Root Complex, either the 8051 or USB Host CPU performs the task of enumerating the PCI Express Ports and all PCI Express devices attached to the downstream Port(s).

The USB 3382 presents one or two downstream PCI Express Ports, selectable by way of the STRAP_PORTCFG input, I^2C, and/or serial EEPROM. When the STRAP_PORTCFG input is pulled or tied Low to Ground, the USB 3382 presents two downstream PCI Express Ports, both with x1 Links, as illustrated in Figure 7-4. When the STRAP_PORTCFG input is pulled High to VDD_IO, the USB 3382 presents a single PCI Express Port x2 Link, as illustrated in Figure 7-5.

Figure 7-4. Root Complex Mode with Two Downstream PCI Express Port x1 Links

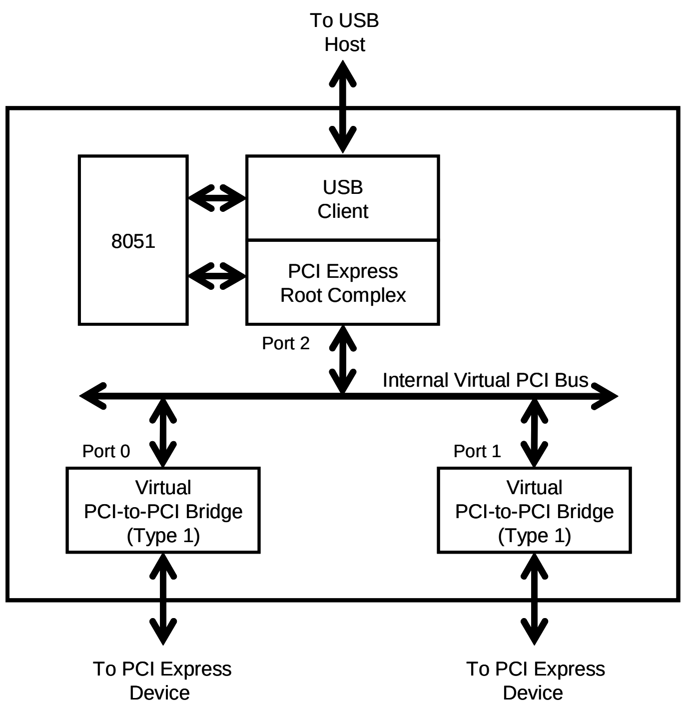
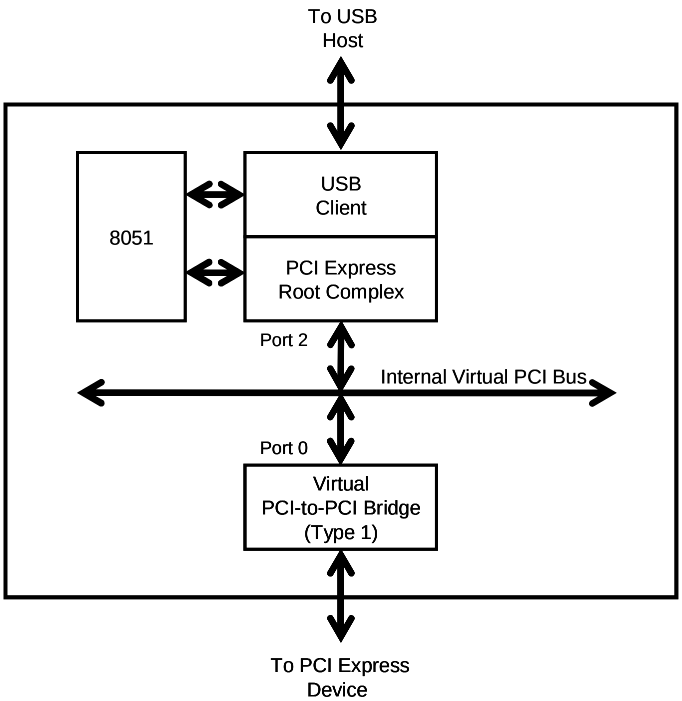


Figure 7-5. Root Complex Mode with Single Downstream PCI Express Port x2 Link



7.4.1 PCI Express Root Complex

The PCI Express Root Complex connects either the 8051 or USB Host CPU to the PCI Express subsystem. Functions in this block include:

- PCI Master Control Registers
- PCI Express Root Port Registers
- Root Complex Event Collector Registers

Each is described in the sections that follow.

7.4.1.1 PCI Master Control Registers

The PCI Master Control registers listed in Table 7-2 are a set of special Cursor registers and associated logic, located in the USB Controller, that allows either the 8051 or USB Host CPU to access PCI Express Space.

Table 7-2. PCI Master Control Registers[a]

Offset	Register	Function
100h	**PCIMSTCTL**	Specifies access type and direction (Read/Write)
104h	**PCIMSTADDR**	Contains the PCI Express address to be accessed
108h	**PCIMSTDATA**	Contains data to be written or data returned from a Read

a. The PCI Master Control register set also includes one Status and one Message register.

Through the PCI Master Control registers, the 8051 or USB Host CPU can generate the following types of accesses into PCI Express space:

- Configuration Read
- Configuration Write
- Memory Read
- Memory Write
- I/O Read
- I/O Write
- PCI Express Messages

PCI Master Control registers are detailed in Section 16.7, “PCI Express/Configuration Cursor Registers.”

Because the 8051 is an 8-bit processor, 8051 accesses to PCI Express Space require a two-step process to transfer up to 32 bits, per transaction. First, the 8051 must use its own set of Cursor registers, located in its dedicated 256-byte Internal Random Access Memory (IRAM) Space, to access the PCI Master Control registers in CSR Space. Then, using the PCI Master Control registers, the 8051 can access PCI Express Space. The 8051 Cursor registers are located at IRAM Addresses F2h through FCh. (Refer to Section 6.3.1, “Configuration Register Access Special Function Registers,” for details.)

USB accesses to PCI Master Control registers are performed by way of the PCIOUT and PCIIN Dedicated endpoints. These endpoints provide direct access from the USB interface to the **PCIMSTCTL** and **PCIMSTADDR** registers. Unlike 8051 accesses into PCI Express Space, which are limited to 32 bits using the **PCIMSTDATA** register, the PCIOUT and PCIIN Dedicated endpoints support Data Payloads up to the programmed PCI Express Maximum Payload Size, either 128 or 256 bytes. Refer to Section 8.6.3, "PCIOUT Endpoint," and Section 8.6.4, "PCIIN Endpoint," for details.

7.4.1.2 PCI Express Root Port Registers

A Root Port is a virtual PCI-to-PCI bridge that connects the Root Complex CPU to the PCI Express hierarchy. Table 7-3 lists the USB Controller registers specific to Root Port support. The PCI Express Root Port registers are detailed in Section 15.10, "PCI Express Capability Registers (Offsets 68h – A0h)."

Table 7-3. PCI Express Root Port Registers

Offset	Register
84h	**Root Capability and Control**
88h	**Root Status**

7.4.1.3 Root Complex Event Collector Registers

The PCI Express Root Port implements the **Advanced Error Reporting (AER) Capability** structure, as defined in the *PCI Express Base r2.1*, Section 7.10. As a Root Port, the **AER** structure includes additional Root Complex Event Collector registers, located in the USB Controller, as listed in Table 7-4. The PCI Express Root Port Event Collector registers are detailed in Section 15.16, "Advanced Error Reporting Extended Capability Registers (Offsets FB4h – FE8h)."

Table 7-4. PCI Express Root Complex Event Collector Registers

Offset	Register
FE0h	**Root Error Command**
FE4h	**Root Error Status**
FE8h	**Error Source ID**

7.4.2 Support for Messages Sent to Root Complex

The Root Complex is the destination for Conventional PCI INT*x*, Error, and PM_PME Messages, as well as MSI/MSI*x* TLPs generated by downstream PCI Express devices. When the Messages and TLPs reach the Root Complex, they are handled in one of two ways:

- If the Messages are for the USB Host CPU, the Message is written to the RCIN FIFO, and an *Interrupt* bit is Set in the STATIN Dedicated endpoint, to notify the USB Host of the Message receipt
- If the Messages are for the 8051, the Message is written to the 8051 Message FIFO, and a Hardware interrupt is generated to the 8051

7.4.3 Power Management Messages

The Root Complex CPU must be able to generate a PME_Turn_Off Message TLP in response to a received PM_PME Message TLP. In the USB 3382, the PCI Master Control registers are used to generate a PME_Turn_Off Message TLP.

7.5 PCI Express Accesses to On-Chip RAM

7.5.1 Legacy Adapter Mode

A PCI Express agent can write to/read from any location in 8051 and/or DMA memory, by sending a Request to an address within the range defined by the **Base Address 1** register *Base Address 1* bit (**BAR1**; USB Controller, offset 14h[31:16]). **BAR1** exposes a 64-KB window to the PCI Express port, but only the lower 32 KB are used to access the 32-KB 8051 and/or DMA memory. PCI Express addresses are mapped into the appropriate 8051 and/or DMA locations within the USB Ingress RAM and PCI Express Ingress RAM, where the 8051 and/or DMA memory is located. Any access outside the 8051 and/or DMA memory range results in an Unsupported Request Response. PCI Express Read Requests to 8051 and/or DMA memory are limited to the PCI Express Maximum Read Request Size, and cannot cross a 4-KB Address Boundary space. PCI Express Write Requests to 8051 and/or DMA memory are limited by the PCI Express Maximum Packet Size.

7.5.2 Enhanced Adapter Mode

A PCI Express agent can write to/read from any location within the 32-KB USB Ingress RAM or 32-KB PCI Express Ingress RAM, by sending a Request to an address within the range defined by the **Base Address 1** register *Base Address 1* bit (**BAR1**; USB Controller, offset 14h[31:16]). **BAR1** exposes a 64-KB window to the PCI Express port. The lower 32 KB access the USB Ingress RAM, and the upper 32 KB access the PCI Express Ingress RAM. PCI Express Read Requests to 8051 and/or DMA memory are limited to the PCI Express Maximum Read Request Size, and cannot cross a 4-KB Address Boundary space. PCI Express Write Requests to 8051 and/or DMA memory are limited by the PCI Express Maximum Packet Size.

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 8 USB Controller Functional Description

8.1 Introduction

The USB Controller joins the *USB r2.0* or *USB r3.0* core to the PCI Express interface. The USB Controller provides the following functions, which are described herein:

- Auto-Enumeration Controller
- FIFO Read/Write pointers
- FIFO Queue Depth monitors
- USB Dedicated Endpoint Controller

8.2 Modes of Operation

Note: *The STRAP_LEGACY input determines whether the USB 3382 operates in Legacy or Enhanced Adapter mode. When STRAP_LEGACY is pulled or tied High to VDD_IO, the USB 3382 operates in Legacy Adapter mode; otherwise, the USB 3382 operates in Enhanced Adapter mode. In both modes, STRAP_RC_MODE input must be pulled or tied Low to Ground.*

8.2.1 Legacy Adapter Mode

In Legacy Adapter mode, the USB 3382 presents a register set, USB endpoints, and FIFO memory that is similar to the PLX NET 2282 PCI to High-Speed USB 2.0 Controller, allowing programs that were written for that platform to be run on the USB 3382, with minimal modification. Legacy Adapter mode fully supports operation at *USB r3.0* data rates.

There are four general-purpose endpoints, which can be either IN or OUT. Each endpoint's direction is controlled by its **EP_CFG** register *Endpoint Direction* bit(s) (USB Controller, offset(s) 300h[7] and 320h, 340h, 360h, 380h[7]). Each endpoint also has its own set of Configuration and Status registers:

- **EP_CFG** register(s) (USB Controller, Legacy Adapter mode, offset(s) 300h, 320h, 340h, 360h, 380h)
- **EP_STAT** register(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch) registers

The endpoint FIFO configuration is determined by the new set of endpoint FIFO registers (not in the NET 2282), as well as the **FIFOCTL** register (USB Controller, offset 38h; included in the NET 2282).

8.2.1.1 Endpoint FIFO Configuration

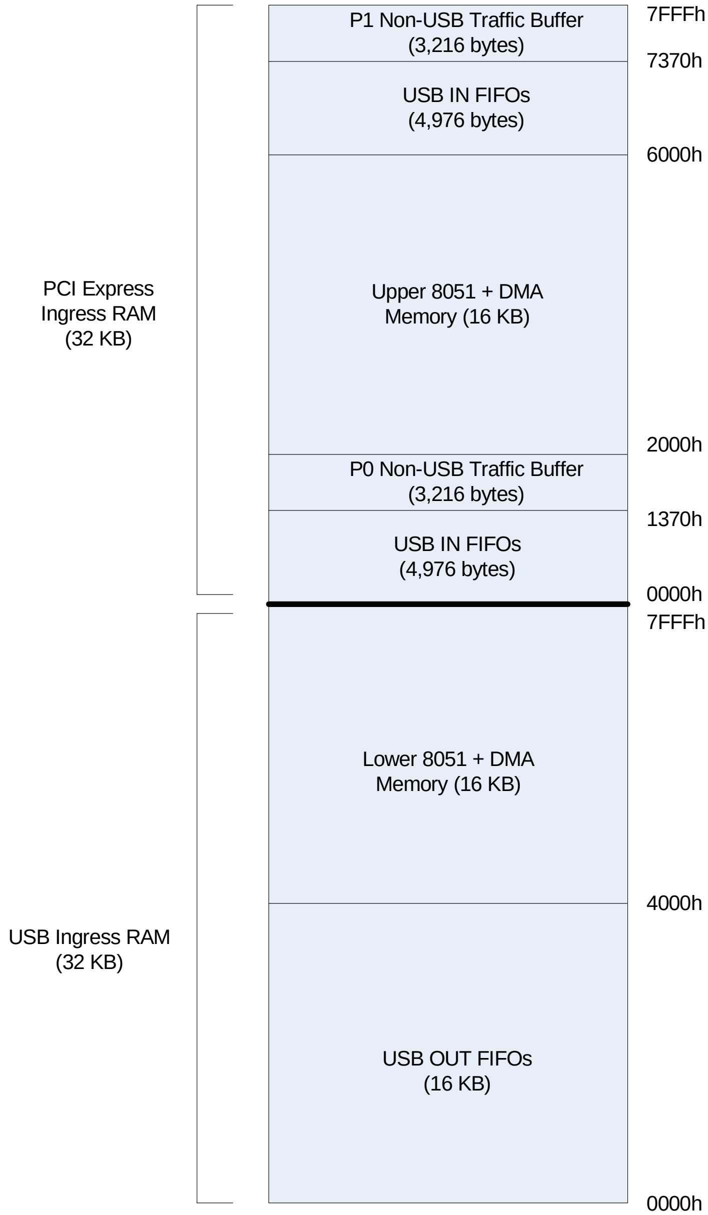
Each USB endpoint has a FIFO associated with it. The IN endpoint FIFOs are located in the PCI Express Ingress RAM. The OUT endpoint FIFOs are located in the USB Ingress RAM.

The USB Ingress RAM provides a total of 16-KB RAM for all OUT endpoints that require FIFO space – EP 0, GPEP[3:0], and PCIOUT. The size of each of these six FIFOs is determined by their respective **EP_FIFO_SIZE_BASE** register *OUT FIFO Size* field(s) (USB Controller, offset(s) 500h, 520h, 540h, 560h, 580h, 5E0h, 600h[2:0]). The size for each FIFO can range from 64 to 4,096 bytes; however, the combined size of the six FIFOs must be less than 16,384 bytes. Each FIFO can be located on any 64-byte RAM boundary. The Base address of each OUT FIFO is determined by the register's *OUT FIFO Base Address* field(s) (field [14:6]), and is relative to the start of the USB Ingress RAM. The Base address resolution is 64 bytes. (For further details regarding OUT FIFOs, refer to Section 8.4.)

The IN FIFOs are placed into two 4,976-byte RAM segments. The following endpoints require an IN FIFO – EP 0, GPEP[3:0], PCIIN, and RCIN. The size of each of these seven FIFOs is determined by their respective **EP_FIFO_SIZE_BASE** register *IN FIFO Size* field(s) (USB Controller, offset(s) 500h, 520h, 540h, 560h, 580h, 5E0h, 600h[18:16]). The size for each FIFO can range from 64 to 4,096 bytes; however, the combined size of the seven FIFOs must be less than 9,952 bytes. There must be 64 bytes of unused memory after each IN FIFO. Each FIFO can be located on any 64-byte boundary, and must fit entirely within one of the two RAM segments. The Base address of each IN FIFO is determined by the register's *IN FIFO Base Address* field(s) (field [30:22]), and is relative to the start of the PCI Express Ingress RAM. The Base address resolution is 64 bytes. (For further details regarding IN FIFOs, refer to Section 8.5.)

Figure 8-1 illustrates the internal RAM layout.

Figure 8-1. Internal RAM Layout



Notes: *The **Base Address 1** register Base Address 1 bit (**BAR1**; USB Controller, offset 14h[31:16]) maps Lower 8051/DMA Memory and Upper 8051+DMA Memory as a single 32-KB block. Other regions of the internal RAM are not accessible from **BAR1**.*

PCI Express Ingress RAM addresses from 2000h through 5FFFh (Upper 8051+DMA Memory) are not available for use as IN FIFO, even when the 8051 and DMA channels are idle.

Typical Endpoint Configurations

Table 8-1 lists typical programmed endpoint FIFO sizes for various endpoints.

Table 8-1. Typical Endpoint Configurations

Type	Endpoint[a]	Adapter/Internal DMA Channel	Root Complex/ External DMA Channel	FIFO Size (in Bytes)	Endpoint Number
Dedicated	CSROUT	***Not used***	OUT	No FIFO	Dh
	CSRIN	***Not used***	IN	No FIFO	Dh
	PCIOUT	***Not used***	OUT	256	Eh
	PCIIN	***Not used***	IN	256	Eh
	STATIN	***Not used***	IN	No FIFO	Fh
	RCIN	***Not used***	IN	512	Ch
General-Purpose	EP 0	IN	IN	512	0
	EP 0	OUT	OUT	512	0
	GPEP0 (OUT)	DMA Channel 0 Write	DMA Channel 0 Read Completion	4K	2
	GPEP0 (IN)	DMA Channel 0 Read	DMA Channel 0 Write Target	2K	2
	GPEP1[b] (OUT)	DMA Channel 1 Write	DMA Channel 0 Descriptor Read Completion	4K (Adapter mode) 256 (Root Complex mode)	4
	GPEP1[b] (IN)	DMA Channel 1 Read	DMA Channel 0 Done Message	2K (Adapter mode) 256 (Root Complex mode)	4
	GPEP2 (OUT)	DMA Channel 2 Write	DMA Channel 1 Read Completion	2K	6
	GPEP2 (IN)	DMA Channel 2 Read	DMA Channel 1 Write Target	2K	6
	GPEP3[b] (OUT)	DMA Channel 3 Write	DMA Channel 1 Descriptor Read Completion	2K (Adapter mode) 256 (Root Complex mode)	8
	GPEP3[b] (IN)	DMA Channel 3 Read	DMA Channel 1 Done Message	2K (Adapter mode) 256 (Root Complex mode)	8

a. *In Legacy Adapter mode, only one direction (OUT or IN) is available for each GPEP endpoint pair. In Enhanced Adapter and Root Complex modes, both directions (OUT and IN) are available for each GPEP endpoint pair.*

b. *In Root Complex mode, if endpoints GPEP1 or GPEP3 are to be used for traffic that uses USB packets larger than 256 bytes, the default FIFO sizes for these endpoints must be increased.*

8.2.2 Enhanced Adapter Mode

In Enhanced Adapter mode, the USB 3382 presents an extended set of USB endpoints designed to operate at SuperSpeed data rates. In this mode, the USB Controller provides four pairs (8 endpoints total) of general-purpose endpoints – GPEP[3:0]. Each GPEP*x* endpoint pair has one OUT and one IN endpoint associated with it. Additionally, each GPEP*x* endpoint pair has one enhanced Endpoint Configuration register and two Response/Status registers associated with it:

- **EP_CFG** register(s) (USB Controller, offset(s) 320h, 340h, 360h, 380h)
- **EP_RSP** register(s) (USB Controller, offset(s) 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h)
- **EP_STAT** register(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch)

The direction of each general-purpose endpoint is fixed, and the **EP_CFG** register *Endpoint Direction* bit(s) (USB Controller, offset(s) 320h, 340h, 360h, 380h[7]) is ignored. The endpoint FIFO configuration is determined by an enhanced set of endpoint FIFO registers.

8.3 Auto-Enumerate Controller

Auto-Enumerate mode, which makes use of the Auto-Enumerate Controller (AEC), relieves the CPU from servicing Standard Read and Write Requests from the USB Host. Each type of Standard Read or Write Request has an associated register bit that determines how the Request is serviced:

- When the bit is Cleared, the Request is passed to the CPU through the Setup registers
- When the bit is Set, the request is serviced without CPU support

The values returned to the Host in Auto-Enumerate mode are determined by the values in other registers, which are discussed in the next section.

8.3.1 Configuration Register Setup

The Configuration registers listed in Table 8-2, located in the USB Controller, determine the auto-enumeration response to USB Standard Requests from EP 0. The default values can be used, or the optional serial EEPROM can load custom values before the enumeration process starts.

Table 8-2. Configuration Registers for Auto-Enumeration Response

Location(s)	Register	Description
Offset 80h	**STDRSP**	Standard Response Control
Offset 84h	**PRODVENDID**	Product and Vendor IDs
Offset 88h	**RELNUM**	Device Release Number
Offset 8Ch	**USBCTL**	USB Control
Offset B4h	**USB_CLASS**	USB Class, Sub-Class, Protocol
Offset B8h	**SS_SEL**	SuperSpeed System Exit Latency
Offset 300h	**EP_CFG** (EP 0 only)	Endpoint Configuration for EP 0
Offsets 304h, 324h/3E4h, 344h/ 404h, 364h/424h, 384h/444h	**EP_RSP**	Endpoint Response for EP 0 and GPEPx
Offsets 320h, 340h, 360h, 380h	**EP_CFG** (GPEP*x* endpoints only)	Endpoint Configuration for GPEPx Endpoints
Index 06h	**HS_MAXPOWER**	High-Speed Maximum Power
Index 07h	**FS_MAXPOWER**	Full-Speed Maximum Power
Index 08h	**HS_INTPOLL_RATE**	High-Speed Interrupt Polling Rate
Index 09h	**FS_INTPOLL_RATE**	Full-Speed Interrupt Polling Rate
Index 0Ah	**HS_NAK_RATE**	High-Speed NAK Rate
Index 20h, 30h, 40h, 50h, 60h, 70h, 80h, 90h	**GPEP[3:0/Out/ In]_HS_MAXPKT**	High-Speed Maximum Packet Size
Index 21h, 31h, 41h, 51h, 61h, 71h, 81h, 91h	**GPEP[3:0/Out/ In]_FS_MAXPKT**	Full-Speed Maximum Packet Size
Index 22h, 32h, 42h, 52h, 62h, 72h, 82h, 92h	**GPEP[3:0/Out/ In]_SS_MAXPKT**	SuperSpeed Maximum Packet Size
Index 84h	**STATIN_HS_INTPOLL_RATE**	High-Speed interrupt polling rate for STATIN
Index 85h	**STATIN_FS_INTPOLL_RATE**	Full-Speed interrupt polling rate for STATIN
Index 86h	**SS_MAXPOWER**	SuperSpeed Maximum Power

8.3.2 Supported Standard Requests

Table 8-3 lists the Standard Requests supported by the AEC.

Table 8-3. AEC-Supported Standard Requests

Standard Request	Recipient	Read/Write	
		USB r2.0	***USB r3.0***
GetStatus(DEVICE)	Device	Read	Read
GetStatus(*ENDPOINT*)	Endpoint	Read	Read
GetStatus(*INTERFACE*)	Interface	Read	Read
SetFeature(ENDPOINT_HALT) ClearFeature(ENDPOINT_HALT)	Endpoint	Write	Write
SetFeature(TEST) ClearFeature(TEST)	Device	Write	–
SetFeature(DEVICE_REMOTE_WAKEUP) ClearFeature(DEVICE_REMOTE_WAKEUP)	Device	Write	–
SetFeature(FUNCTION_SUSPEND) ClearFeature(FUNCTION_SUSPEND)	Interface	–	Write
SetFeature(U1_ENABLE) ClearFeature(U1_ENABLE)	Device	–	Write
SetFeature(U2_ENABLE) ClearFeature(U2_ENABLE)	Device	–	Write
SetFeature(LTM_ENABLE) ClearFeature(LTM_ENABLE)	Device	–	Write
GetDescriptor(DEVICE)	Device	Read	Read
GetDescriptor(CONFIGURATION) (includes Interfaces, Endpoints)	Device	Read	Read
GetDescriptor(SUPERSPEED_USB_ENDPOINT_ COMPANION) (provided only at *USB r3.0* SuperSpeed)	Device	–	Read
GetDescriptor(LANGID) (String0)	Device	Read	Read
GetDescriptor(MANUFACTURER) (String1)	Device	Read	Read
GetDescriptor(PRODUCT) (String2)	Device	Read	Read
GetDescriptor(SERIAL_NUMBER) (String3)	Device	Read	Read
GetDescriptor(DEVICE_QUALIFIER) (supported only at *USB r2.0* speeds)	Device	Read	–
GetDescriptor(OTHER_SPEED_CONFIGURATION) (supported only at *USB r2.0* speeds)	Device	Read	–
GetDescriptor(BOS) (*USB r2.0* Extension; SuperSpeed Device Capability; provided at all USB speeds)	Device	Read	Read
SetAddress	Device	Write	Write
SetConfiguration	Device	Write	Write

Table 8-3. AEC-Supported Standard Requests *(Cont.)*

Standard Request	Recipient	Read/Write	
		USB r2.0	*USB r3.0*
GetConfiguration	Device	Read	Read
SetInterface (Supports only Interface 0)	Interface	Write	Write
GetInterface	Interface	Read	Read
SetSel (System Exit Latency)	Device	–	Write
SetIsochDelay (Isochronous Delay)	Device	–	Write

8.3.3 Control Transfer Detection

The AEC monitors the *USB r2.0* or *USB r3.0* core for the arrival of a valid Setup packet. Byte 0 of the 8-byte Setup packet is the *bmRequestType*, which determines the transfer characteristics, as listed in Table 8-4. The remaining seven bytes of the Setup packet are described in Table 8-5.

The AEC supports only Control transfers of type Standard. All other types – Class and Vendor – must be handled by the local CPU.

Table 8-4. Setup Packet, Byte 0 – *bmRequestType*

Bit(s)	Characteristic
4:0	**Recipient** 00h = Device 01h = Interface 02h = Endpoint 03h = Other All other encodings are ***reserved***.
6:5	**Type** 00b = Standard 01b = Class 10b = Vendor 11b = ***Reserved***
7	**Data Transfer Direction** 0 = Host to device (Control Write) 1 = Device to Host (Control Read)

Table 8-5. Standard Request Setup Packet, Bytes 1 through 7

Byte	Setup Packet	Function
1	*bRequest*	Type of Standard Request
2, 3	*wValue*	Request-specific parameter
4, 5	*wIndex*	Request-specific parameter
6, 7	*wLength*	Length of Data stage

8.3.3.1 Standard Control Write Requests

When a Standard Control Write is detected, the *bmRequestType* and *bRequest* bytes (Bytes 0 and 1, respectively) are examined to determine the type of Control Write. Each Standard Control Write type has an *AEC Enable* bit. The AEC begins processing an action for a Control Write, if the corresponding *AEC Enable* bit is Set. For each Request, a Write Request is made to the appropriate register. The Write Data Bus is based upon data from the Setup packet. When the Write is complete, the Request is removed, and the Control Write Status stage is ACKed.

The AEC supports the Standard Control Write Requests listed in Table 8-6.

Table 8-6. AEC-Supported Standard Control Write Requests

Standard Write Request	Recipient	*bRequest*	*wValue*	*wIndex*	*wLength*
SetFeature(ENDPOINT_HALT) ClearFeature(ENDPOINT_HALT)	Endpoint	SET_FEATURE/ CLEAR_FEATURE	Feature Selector 0	Endpoint Number	0
SetFeature(DEVICE_REMOTE_WAKEUP) ClearFeature(DEVICE_REMOTE_WAKEUP)	Device	SET_FEATURE/ CLEAR_FEATURE	Feature Selector 1	0	0
SetFeature(TEST) ClearFeature(TEST)	Device	SET_FEATURE/ CLEAR_FEATURE	Feature Selector 2	Test Selector	0
SetFeature(FUNCTION_SUSPEND) ClearFeature(FUNCTION_SUSPEND)	Interface	SET_FEATURE/ CLEAR_FEATURE	Feature Selector 0	{Suspend Option, Interface Number}	0
SetFeature(U1_ENABLE) ClearFeature(U1_ENABLE)	Device	SET_FEATURE/ CLEAR_FEATURE	Feature Selector 48	0	0
SetFeature(U2_ENABLE) ClearFeature(U2_ENABLE)	Device	SET_FEATURE/ CLEAR_FEATURE	Feature Selector 49	0	0
SetFeature(LTM_ENABLE) ClearFeature(LTM_ENABLE)	Device	SET_FEATURE/ CLEAR_FEATURE	Feature Selector 50	0	0
SetAddress	Device	SET_ADDRESS	Device Address	0	0
SetConfiguration	Device	SET_CONFIGURATION	Configuration Value	0	0
SetInterface	Interface	SET_INTERFACE	Alternate Setting	Interface	0
SetSel (System Exit Latency)	Device	SET_SEL	0	0	0
SetIsochDelay (Isochronous Delay)	Device	SET_ISOCH_DELAY	Delay Value	0	0

8.3.3.2 Standard Control Read Requests

When a Standard Control Read is detected, the *bmRequestType* and *bRequest* bytes (Bytes 0 and 1, respectively) are examined to determine the type of Control Read. Each Standard Control Read type has an *AEC Enable* bit. The AEC begins processing a Control Read, if the corresponding *AEC Enable* bit is Set. For each Request, a response is written to the EP 0 IN FIFO, using the low-priority PCI Express Ingress RAM Bus. The response packet is returned to the Host, during the Control Read Data stage.

For Descriptors, the upper byte of *wValue* (Byte 3) is the Descriptor type, and the lower byte (Byte 2) is the index. The index is used only for Configuration and String Descriptors.

The Configuration Descriptor size is based upon the quantity of enabled endpoints and Link speed. A state machine determines the quantity of bytes to be written, then writes them to the EP 0 FIFO.

The AEC supports the Standard Control Read Requests listed in Table 8-7.

Table 8-7. AEC-Supported Standard Control Read Requests

Standard Read Request	Recipient	*bRequest*	*wValue*[a]	*wIndex*	*wLength*
GetStatus(DEVICE)	Device	GET_STATUS	0	0	2
GetStatus(*INTERFACE*)	Interface	GET_STATUS	0	Interface	2
GetStatus(*ENDPOINT*)	Endpoint	GET_STATUS	0	Endpoint	2
GetDescriptor(DEVICE)	Device	GET_DESCRIPTOR	{1,0}	0	18
GetDescriptor(CONFIGURATION) (includes Interfaces, Endpoints)	Device	GET_DESCRIPTOR	{2,0}	0	Varies
GetDescriptor(SUPERSPEED_USB_ ENDPOINT_COMPANION) (provided only at *USB r3.0* SuperSpeed)	Device	GET_DESCRIPTOR	{48,0}	0	6
GetDescriptor(LANGID) (String0)	Device	GET_DESCRIPTOR	{3,0}	LangID	4
GetDescriptor(MANUFACTURER) (String1)	Device	GET_DESCRIPTOR	{3,1}	LangID	42
GetDescriptor(PRODUCT) (String2)	Device	GET_DESCRIPTOR	{3,2}	LangID	Varies
GetDescriptor(SERIAL_NUMBER) (String3)	Device	GET_DESCRIPTOR	{3,3}	LangID	22
GetDescriptor(DEVICE_QUALIFIER) (supported only at *USB r2.0* speeds)	Device	GET_DESCRIPTOR	{6,0}	0	10
GetDescriptor(OTHER_SPEED_ CONFIGURATION) (supported only at *USB r2.0* speeds)	Device	GET_DESCRIPTOR	{7,0}	0	9
GetDescriptor(BOS) (*USB r2.0* Extension; SuperSpeed Device Capability; provided at all USB speeds)	Device	GET_DESCRIPTOR	{15,0}	0	22
GetConfiguration	Device	GET_CONFIGURATION	0	0	1
GetInterface	Interface	GET_INTERFACE	0	Interface	1

a. The upper byte of wValue (Byte 3) is the Descriptor type. The lower byte (Byte 2) is the index.

8.4 USB OUT FIFO Controller

The USB OUT FIFO Controller manages the flow of data between the *USB r2.0* or *USB r3.0* core and the USB endpoint FIFO areas within the USB Ingress RAM.

8.4.1 OUT FIFO Writes

The USB OUT FIFO Controller accepts packets received by the *USB r2.0* or *USB r3.0* core, then writes the data received to the appropriate endpoint FIFO space within the USB Ingress RAM. Each OUT endpoint is assigned a region of memory, based upon the **EP_FIFO_SIZE_BASE** register *IN FIFO Base Address*, *OUT FIFO Base Address*, *IN FIFO Size*, and *OUT FIFO Size* fields (USB Controller, offset(s) 500h, 520h, 540h, 560h, 580h, 5E0h, 600h[30:22, 14:6, 18:16, and 2:0], respectively). OUT packets for EP 0 and GPEP[3:0] are stored in their respective OUT FIFOs. OUT packets to CSROUT are intercepted and stored in internal Configuration registers, rather than written to a FIFO. The first 2 DWords of OUT packets to the PCIOUT endpoint FIFO are stored in the **PCIMSTCTL** and **PCIMSTADDR** registers, and the remainder are written to the PCI Express interface.

When receiving data, the USB 3382 NAKs the Host (indicating that the USB 3382 cannot accept the data) when one of the following conditions is met:

- Endpoint's FIFO is full, –or–
- Endpoint's **EP_RSP** register *NAK OUT Packets Mode* and *NAK Packets* bits are Set (USB Controller, offset(s) 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h)

If the packets received are of maximum size, then additional packets are received, independent of the *NAK OUT Packets Mode* bit state. This bit causes additional OUT packets to be NAKed if the last packet received was a short packet. If the *NAK OUT Packets Mode* bit is Set (Blocking mode), USB OUT transfers can overlap with the CPU (8051 or PCI Express Host), unloading the data in the following sequence:

1. CPU responds to the **EP_STAT** register *Data Packet Received Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[3]), then reads the **EP_AVAIL** register *Endpoint Available Counter* field(s) (USB Controller, offset(s) 310h, 330h/3F0h, 350h/410h, 370h/430h, 390h/450h[13:0]) to determine the quantity of bytes in the current packet.
2. CPU Clears the *Data Packet Received Interrupt* and *NAK Packets* bits, allowing the next packet to be received.
3. CPU can now unload data from the FIFO during the next USB OUT transaction.

If the *NAK OUT Packets Mode* bit is Cleared (Non-Blocking mode), the USB 3382 accepts packets as long as there is sufficient space for the complete packet in the FIFO. There are no indications of packet boundaries when the FIFO consists of multiple packets.

8.4.2 OUT FIFO Reads

Data can be read from the OUT endpoint FIFOs, using any of the following methods:

- PCI Express Memory Read Request targeting the endpoint FIFOs, by way of the endpoint's **EP_FIFO_SIZE_BASE** register(s) (USB Controller, offset(s) 500h, 520h, 540h, 560h, 580h, 5E0h, 600h)
- DMA channels can be used to read data from the OUT endpoint FIFOs, and write that data to PCI Express (this is the preferred method for High-Speed transfer between USB FIFOs and PCI Express)
- Register accesses to the **EP_DATA** and **EP_DATA1** registers (USB Controller, offset(s) 314h, 334h/3F4h, 354h/414h, 374h/434h, 394h/454h and 318h, 338h/3F8h, 358h/418h, 378h/438h, 398h/458h, respectively)
- AEC reads Standard Write data

8.4.3 OUT FIFO Flush

The following events result in an OUT FIFO flush:

- Write to the endpoint's **EP_STAT** register *FIFO Flush* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[9])
- PCI Express interface is reset
- Setup packet is ACKed (EP 0 only)
- Incorrect packet length (PCIOUT endpoint only)

8.4.4 OUT Endpoint Halt Conditions

OUT endpoints can be halted as a result of the following conditions:

- Write to the endpoint's **EP_RSP** register *Endpoint Halt Set* bit(s) (USB Controller, offset(s) 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h[8]), to explicitly Set the bit
- SetFeature(ENDPOINT_HALT) Request that is handled by the Auto-Enumerate Controller
- ECC error was detected during an Endpoint FIFO Read

8.5 USB IN FIFO Controller

The USB IN FIFO Controller manages the flow of data between USB endpoint FIFOs in the PCI Express Ingress RAM and the *USB r2.0* or *USB r3.0* core for IN endpoints.

8.5.1 IN FIFO Writes

Data bound for the USB Bus can arrive at the USB IN FIFOs, by way of the following methods:

- DMA Reads, from the PCI Express interface (preferred method for high-bandwidth transfers)
- Memory Writes, from the PCI Express interface, targeting the **BAR2** and/or **BAR3** IN FIFO (**EP_FIFO_SIZE_BASE** register(s) (USB Controller, offset(s) 500h, 520h, 540h, 560h, 580h, 5E0h, 600h))
- Register accesses (Writes), to the **EP_DATA** register(s) (USB Controller, offset(s) 314h, 334h/3F4h, 354h/414h, 374h/434h, 394h/454h)
- AEC Auto-Enumerate Response data

Note: ***BAR3*** *is available only in Enhanced Adapter mode.*

In Legacy Adapter mode, the **FIFOCTL** register *PCI BAR2 Select* bit (USB Controller, offset 38h[2]) determines how **BAR2** is mapped to the GPEP*x* FIFOs:

- *PCI BAR2 Select* bit is Cleared
  - **BAR2** Quadrant 0 = Writes to/reads from GPEP0 (writes for IN endpoint, reads from OUT endpoint)
  - **BAR2** Quadrant 1 = Writes to/reads from GPEP1 (writes for IN endpoint, reads from OUT endpoint)
  - **BAR2** Quadrant 2 = Writes to/reads from GPEP2 (writes for IN endpoint, reads from OUT endpoint)
  - **BAR2** Quadrant 3 = Writes to/reads from GPEP3 (writes for IN endpoint, reads from OUT endpoint)
- *PCI BAR2 Select* bit is Set
  - **BAR2** first half = Writes to GPEP0
  - **BAR2** first half = Reads from GPEP1
  - **BAR2** second half = Writes to GPEP2
  - **BAR2** second half = Reads from GPEP3

In Enhanced Adapter mode, the **BAR2CTL** and **BAR2CTL** registers (USB Controller, offsets 3Ch and 40h, respectively) determines how **BAR2** and **BAR3** are mapped to the GPEP*x* FIFOs:

- **BAR2CTL** register allows any GPEP*x* FIFO to be mapped into any one of the four quadrants of the **BAR2** space
- **BAR3CTL** register allows any GPEP*x* FIFO to be mapped into any one of the four quadrants of the **BAR3** space

8.5.1.1 PCI Express Credits

The **INCH Status Control for Ports 0 and 1** register *INCH FIFO OFF* field (Port 0, offset 9F0h[31:28]) applies to the four GPEP IN endpoints when they are written through **BAR2** or **BAR3** from the PCI Express interface:

- When one of the bits is Cleared, its corresponding GPEP IN FIFO is included in the PCI Express credit advertisement calculations
- When one of the bits is Set, its corresponding GPEP IN FIFO is ***not*** included in the PCI Express credit advertisement calculations

The quantity of credits advertised is determined by the FIFO with the least amount of available space. A GPEP IN endpoint is typically excluded from credit calculations when its corresponding FIFO is small. If this small FIFO were included, very few Posted credits could be advertised, and the larger endpoints would suffer a performance penalty. Setting one or more of the bits in the *INCH FIFO OFF* field resolves this problem.

In Root Complex mode, the GPEP1 and GPEP3 IN FIFOs default to the smallest FIFO size (256 bytes). Therefore, by default, the GPEP1 and GPEP3 endpoints are not included in the credit calculations for Root Complex mode.

Note: ***BAR3*** *is available only in Enhanced Adapter mode.*

8.5.2 FIFO Reads from *USB r2.0* or *USB r3.0* Core

The USB Controller reads data from the IN FIFOs, then writes data to the *USB r2.0* or *USB r3.0* core for transmission to the USB Host. When an IN token arrives and the corresponding IN FIFO has a validated packet, the *USB r2.0* or *USB r3.0* core starts issuing Read Requests.

The FIFO Read pointers are initialized to the FIFO Base Address. USB packets are always aligned on 2-DWord boundaries; therefore, the pointers represent a 2-DWord address, rather than a Byte address.

8.5.3 IN FIFO (Write Target) Packet Validation

If an IN FIFO does not have validated data, the USB 3382 responds with a NAK/NRDY to the USB Host. There are several methods for validating the IN FIFO data:

- For large amounts of data, the PCI Express agent can write data to the IN FIFO, as long as FIFO space is available. When there is at least the specified quantity of **EP_*n*_MAXPKT** bytes in the FIFO, the USB 3382 responds to an IN token with a Data packet. If the entire Data transfer is a multiple of the specified quantity of **EP_*n*_MAXPKT** bytes, no other action is required to validate the FIFO data. If a zero-length packet (ZLP) must be sent to the Host, the CPU can Clear the endpoint's **EP_CFG** register *EP FIFO Byte Count* field(s) (USB Controller, offset(s) 300h[18:16] and 320h, 340h, 360h, 380h[18:16]), then write to the **EP_DATA** register(s) (USB Controller, offset(s) 314h, 334h/3F4h, 354h/414h, 374h/434h, 394h/454h).

  This feature is enabled only if the endpoint's **EP_CFG** register *Byte Packing Enable* bit(s) (USB Controller, offset(s) 300h[11] and 320h, 340h, 360h, 380h[11]) is not Set.
- When using the internal DMA Controller, the DMA Byte Count is used. This Counter is initialized to the total Transfer Byte Count, before data is written to the IN FIFO. The Counter decrements as data is written to the FIFO. When the Counter reaches 0, the remaining data in the FIFO is validated. Excess bytes in the last word are automatically ignored. If the last packet of a transfer has the specified quantity of **EP_*n*_MAXPKT** bytes (quantity indicated by the register value), the USB 3382 responds to the next IN token with a ZLP.
- If the upper bits of the PCI Express packet Byte Enables are not Set, this is considered a non-contiguous Write, and results in a short packet validation if the **EP_CFG** register *Byte Packing Enable* bit(s) is not Set.
- The **EP_STAT** register *FIFO Validate* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[8]) validates data in the IN FIFO:
  - If the last word written to the FIFO did not contain contiguous Byte Enables that are Set, or if the *EP FIFO Byte Count* field(s) is not a value of 4 (partial Byte Enables), writing to the *FIFO Validate* bit(s) has no effect. Writing to the *FIFO Validate* bit(s) a second time causes a ZLP to be placed into the FIFO.
  - If the last word written to the FIFO has all the Byte Enables Set, writing to the *FIFO Validate* bit(s) validates the data if the Byte Count is less than the programmed Maximum Packet Size, or causes a ZLP to be written if there is an even multiple of Maximum Packet Size bytes in the FIFO (*that is,* no short packet).

    USB protocol requires that a ZLP to be appended when all packets within the transfer are of Maximum Packet Size. The FIFO Validate function takes this into account.
- Writing any data value to the GPEP*x* **EP_VAL** register(s) (USB Controller, offset(s) 33Ch, 35Ch, 37Ch, 39Ch) performs the same function as the *FIFO Validate* bit(s) described in the previous bullet.

8.5.4 IN FIFO Flush

The following events result in an IN FIFO flush:

- Write to the endpoint's **EP_STAT** register *FIFO Flush* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[9])
- Endpoint is halted
- USB Host is reset
- IN endpoint PCI Express credit timeout
- Setup packet is ACKed (EP 0 only)
  - New OUT packet sent to PCIOUT, –or–
  - Malformed Completion TLP is received, –or–
  - Completion timeout for PCI Express read (PCIIN endpoint only)

8.5.5 PCI Express Credit Timeout

A deadlock can occur when a GPEP IN FIFO has filled up and no USB forward progress is being made. This can occur if there is a problem with the USB Host. In this situation, the USB 3382 is no longer able to update PCI Express credits, thus preventing the PCI Express Host from performing Write transactions to the device. To prevent such occurrences, there is a programmable Timer for each GPEP that runs when either of the following conditions are met:

- Corresponding IN FIFO is full
- PCI Express Port is unable to update credits

When the timer times out, the corresponding GPEP IN FIFO is flushed, a *Status* bit is set, and an interrupt is generated. The **IN_TIMEOUT** register (USB Controller, offset CCh) supports this feature.

8.5.6 IN Endpoint Halt Conditions

IN endpoints can be halted as a result of the following conditions:

- Firmware writes to the endpoint's **EP_RSP** register *Endpoint Halt Set* bit(s) (USB Controller, offset(s) 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h[8]), to explicitly Set the bit
- SetFeature(ENDPOINT_HALT) Request that is handled by the Auto-Enumerate Controller
- ECC error is detected during an Endpoint FIFO Read
- PCI Express error incurred during a PCI Express write to the **BAR2** and/or **BAR3** IN FIFO
  - TLP has a malformed Header
  - TLP *EP* (*Poison*) bit is Set
  - TLP has an invalid Virtual Channel specified
  - Destination is an OUT GPEP (wrong direction)
  - Device is placed into the PCI Express PCI-PM D3hot state
  - USB Controller or in-line internal Port (PCI-to-PCI bridge) has the **PCI Command** register *Memory Access Enable* bit (USB Controller, offset 04h[1]) or **PCI Command** register *Memory Access Enable* bit (All Ports, offset 04h[1]) Cleared
  - Enhanced Adapter Mode Only
    - In-line internal Port (PCI-to-PCI bridge) **Bridge Control** register *Secondary Bus Reset* bit (All Ports, offset 3Ch[22]) is Set
    - Access Control Services (ACS) error
- DMA Controller Aborts during Read from the PCI Express interface
  - Completion has a malformed Header
  - Completion has the wrong payload length
  - Completion *EP* (*Poison*) bit is Set
  - Read Completion timeout

8.6 Dedicated Endpoints

The USB 3382 supports the six Dedicated endpoints listed in Table 8-8. Each is discussed in the sections that follow.

Note: *The Cursor registers (refer to Section 16.7, "PCI Express/Configuration Cursor Registers") used by the Dedicated Endpoints are also used by the internal 8051. No hardware locking is provided between 8051 and USB accesses to these Cursor registers; therefore, firmware must provide this function if these two devices are attempting simultaneous accesses. The **SEMAPHORE** register Semaphore bit (USB Controller, offset 118h[0]) can be used for this resource locking. The Requester reads the Semaphore bit until it receives a value of 0. The Semaphore bit is automatically Set, and the Requester is granted permission to access the Cursor registers. When the Requester finishes accessing the Cursor registers, it writes a 0, to Clear the Semaphore bit.*

Table 8-8. Dedicated Endpoints

Endpoint	Type	Description
CSROUT	Bulk	Initiates register Write and Read Requests
CSRIN		Collects register Read Completions
PCIOUT		Initiates PCI Express Write and Read Requests
PCIIN		Collects PCI Express Read Completions
STATIN (Root Complex mode only)	Bulk or Interrupt	Provides interrupt status to the USB Host
RCIN (Root Complex mode only)	Interrupt	Provides MSI and Power Management Messages to the USB Host

8.6.1 CSROUT Dedicated Endpoint

CSROUT is a Bulk endpoint that allows a USB Host to access the on-chip registers. The endpoint is used to initiate both Reads and Writes. Packets sent to this endpoint consist of the format listed in Table 8-9.

The 16-bit Register/Memory Address field selects one of the internal registers, or a location within the 8051 Program RAM. Because the registers are DWord-aligned, the two least significant bits of the address are not used for register accesses. The **CSRCTL** register *CSR Space Select* field (USB Controller, offset 110h[5:4]) determines whether the PCI Express Configuration registers (offsets 00h to FFh), Memory-Mapped Configuration registers (offsets 000h to 7FFh), or 8051/DMA Descriptor RAM are accessed, as listed in Table 8-10. The Byte Enables determine which of the four bytes within the selected register are accessed (Byte Enable 0 corresponds to bits [7:0], and so forth).

Table 8-9. CSROUT Endpoint Packet Format

<table>
<tr><th rowspan="2">Byte Index</th><th colspan="2">Destination Register Bytes</th></tr>
<tr><th>Register</th><th>Bits</th></tr>
<tr><td>0</td><td rowspan="4">CSRCTL register
(USB Controller, offset 110h)</td><td>[7:0]</td></tr>
<tr><td>1</td><td>[15:8]</td></tr>
<tr><td>2</td><td>[23:16]</td></tr>
<tr><td>3</td><td>[31:24]</td></tr>
<tr><td>4</td><td rowspan="4">CSRDATA register
(USB Controller, offset 114h)</td><td>[7:0]</td></tr>
<tr><td>5</td><td>[15:8]</td></tr>
<tr><td>6</td><td>[23:16]</td></tr>
<tr><td>7</td><td>[31:24]</td></tr>
</table>

Table 8-10. CSROUT Address Space Selects

<table>
<tr><th>CSRCTL Register
CSR Space Select
Field Value</th><th colspan="2">Resource</th></tr>
<tr><td rowspan="6">00b</td><td colspan="2">PCI Express Configuration registers (Enhanced Adapter mode only)</td></tr>
<tr><td>Address</td><td>Description</td></tr>
<tr><td>0XXXh</td><td>Port 0 Type 1 Configuration registers</td></tr>
<tr><td>1XXXh</td><td>Port 1 Type 1 Configuration registers</td></tr>
<tr><td>2XXXh</td><td>Port 2 Type 1 Configuration registers</td></tr>
<tr><td>3XXXh</td><td>USB Controller Type 0 Configuration registers</td></tr>
<tr><td>01b</td><td colspan="2">USB Controller Configuration registers</td></tr>
<tr><td>10b</td><td colspan="2">8051/DMA Descriptor RAM</td></tr>
<tr><td>11b</td><td colspan="2">Reserved</td></tr>
</table>

8.6.1.1 CSR Cursor Register Write

When the CSROUT Dedicated endpoint receives a valid OUT packet, the Payload data containing Control, Address, and optional Write data is written to the **CSRCTL** and **CSRDATA** Cursor registers (for Writes), as listed in Table 8-9. From the Cursor registers, the appropriate Write or Read operation to/from the designated register follows.

If the CSROUT packet is received without an error, the 2 DWords are directly written to the Configuration Cursor registers. Subsequent OUT packets to this endpoint are NAKed/NRDYed, until the Write or Read completes.

For a Configuration Register Read, CSROUT bytes past the fourth byte are ignored. For a Configuration Register Write, CSROUT bytes past the eighth byte are ignored.

8.6.1.2 CSROUT STALL Conditions

OUT packets to the CSROUT Dedicated endpoint result in a STALL and endpoint Halt, for the following conditions:

- USB packet size < 4 bytes
- USB packet size < 8 bytes for a Write
- USB packet size is not a multiple of 4 bytes

8.6.1.3 CSROUT Latency

In USB High-Speed mode, the delay from the beginning of the OUT token until the register Write transaction finishes and the OUT packet is ACKed is approximately 900 ns. In USB Full-Speed mode, the delay is approximately 14.5 µs. Taking into account the Host latencies on a typical PC platform, the average time to perform a Write is approximately 250 µs in High-Speed mode and 8.5 ms in Full-Speed mode.

8.6.1.4 CSROUT Flow Control

For *USB r2.0*, a NYET handshake is returned to the Host for each CSROUT packet, indicating that the packet was accepted, but to not send any more packets. The Host pings the CSROUT Dedicated endpoint. When the CSR transaction completes, an ACK is returned in response to a PING. The Host can then send a new packet to CSROUT.

For *USB r3.0*, an ACK with *NumP*=0 is returned to the Host for each CSROUT packet, indicating that the packet was accepted, but to not send any more packets. The CSROUT endpoint is now in the Flow Control state. When the CSR transaction completes, an ERDY packet is sent to the Host, taking the CSROUT Dedicated endpoint out of the Flow Control state. The Host can then send a new packet to CSROUT.

8.6.2 CSRIN Endpoint

CSRIN is a Bulk endpoint that provides CSR Read data to a USB Host. Packets read from this endpoint consist of four bytes, formatted as listed in Table 8-11.

This endpoint returns the current value of the **CSRDATA** Cursor register (USB Controller, offset 114h), in response to an IN token. The **CSRDATA** register is valid if a previous CSROUT packet initiated a CSR Read. When the IN packet is ACKed, the value in the **CSRDATA** register is invalidated. The CSRIN endpoint returns a NAK/NRDY handshake if there is no valid data in the **CSRDATA** register.

Table 8-11. CSRIN Endpoint Packet Format

Byte Index	Description
0	Register Read Data 0 (LSB)
1	Register Read Data 1
2	Register Read Data 2
3	Register Read Data 3 (MSB)

8.6.2.1 CSRIN Latency

In USB High-Speed mode, the delay from the beginning of the OUT token (for the OUT packet that Sets the CSR Read address) until the Data IN packet (containing the CSR Read data) is ACKed is approximately 2.5 µs. In USB Full-Speed mode, the delay is approximately 28 µs.

Taking into account the Host latencies on a typical PC platform, the average time to perform a CSR Read in High-Speed mode is approximately 500 µs (250 µs to Set the address, and 250 µs to read the data). In Full-Speed mode, the average time to perform a CSR Read is approximately 17 ms (8.5 ms to Set the address, and 8.5 ms to read the data).

8.6.3 PCIOUT Endpoint

PCIOUT is a Bulk endpoint that allows the USB Host to initiate Read and Write Requests to PCI Express Space, using the PCI Master Control Cursor registers. Packets sent to this endpoint consist of the format listed in Table 8-12.

There can be from 0 to 64 Payload DWords, requiring USB packet sizes from 8 to 264 bytes.

Table 8-12. PCIOUT Packet Format

Byte Index	Destination Register Bytes	
	Register	Bits
0	**PCIMSTCTL** register (USB Controller, offset 100h)	[7:0]
1		[15:8]
2		[23:16]
3		[31:24]
4	**PCIMSTADDR** register (USB Controller, offset 104h)	[7:0]
5		[15:8]
6		[23:16]
7		[31:24]
8 through 11	–	Payload DW0 (LSB first; to PCIOUT FIFO)
12 through 15	–	Payload DW1 (LSB first; to PCIOUT FIFO)
…	–	And so forth

8.6.3.1 PCIOUT USB Handshake Responses

Table 8-13 lists the PCIOUT endpoint responses in Full-Speed, High-Speed, and SuperSpeed modes.

Table 8-13. PCIOUT USB Handshake Responses

PCI Express Request Type	Full-Speed Mode	High-Speed Mode	SuperSpeed Mode
Memory Write	ACK packets until request length is fulfilled, then NAK subsequent packets until a PCI Express ACK is received and the OUT FIFO is empty	NYET packet, NAK PINGs until a PCI Express ACK is received and the OUT FIFO is empty, then ACK the PING	ACK with *NumP*=0, send ERDY when a PCI Express ACK is received and the OUT FIFO is empty
Memory Read	ACK packet, then NAK subsequent packets until a PCI Express Completion is received	NYET packet, NAK PINGs until a PCI Express Completion is received, then ACK the PING	ACK with *NumP*=0, send ERDY when a PCI Express Completion is received
I/O Read			
Configuration Read			
I/O Write	NAK packet, but capture payload; continue to NAK until a PCI Express Completion is received, then ACK and discard the packet	NAK packet, but capture payload; continue to NAK until a PCI Express Completion is received, then ACK and discard the packet	ACK with *NumP*=0, send ERDY when a PCI Express Completion is received and the Reader has de-allocated the PCI Express Write
Configuration Write			

8.6.3.2 Non-Contiguous Byte Enables

Non-contiguous first Byte Enables are allowed only for 1-DWord Payloads. Multiple DWord Payloads must have contiguous first Byte Enables. Contiguous Byte Enables do not have any Cleared Byte Enables after the first active Byte Enable. *For example*, 1101b is non-contiguous, but 1110b is contiguous.

For a 1-DWord Read with non-contiguous Byte Enables, the first valid byte and all other most significant bytes are returned to the Host. *For example*, if the Byte Enables are 1010b, Bytes 1, 2, and 3 are returned. For a multiple-DWord Read, the Byte Enables must be contiguous.

For the first DWord, the first valid byte and all other most significant bytes are returned to the Host. *For example*, if the Byte Enables are 1100b, Bytes 2 and 3 are returned to the Host, followed by data from subsequent DWords.

8.6.3.3 Configuration Address Format

Table 8-14 lists the **PCIMSTADDR** register (USB Controller, offset 104h) format for PCI Express Configuration transactions.

Table 8-14. PCIMSTADDR Register Format for PCI Express Configuration Transactions

Bit(s)	Description
7:0	Bus Number
10:8	Function Number
15:11	Device Number
19:16	Extended Register Number
23:20	***Reserved***
24	Configuration Transaction Type 0 = Type 0 1 = Type 1
25	***Reserved***
31:26	Register Number

8.6.3.4 Message Format

Table 8-15 lists the **PCIMSTADDR** register (USB Controller, offset 104h) format for PCI Express Messages.

The Bus Number, Function Number, and Device Number are used only for Messages that are routed by ID. The Vendor ID is used for all types of Vendor-Defined Messages.

Table 8-15. PCIMSTADDR Register Format for PCI Express Messages

Bit(s)	Description
7:0	Bus Number
10:8	Function Number
15:11	Device Number
23:16	Vendor ID[15:8]
31:24	Vendor ID[7:0]

8.6.3.5 Non-Posted Writes

For I/O and Configuration Writes, the PCIOUT Dedicated endpoint NAKs subsequent Requests until a Completion is received on the PCI Express interface. The Completion does not cause any data to be stored in the PCIIN endpoint FIFO.

8.6.3.6 PCI Master Control Cursor Register Write

When the PCI Master Control Cursor registers are written and the **PCIMSTCTL** register *PCI Express Master Start* bit (USB Controller, offset 100h[6]) is Set, a PCI Express Request TLP is generated to PCI Express Space. For PCI Express Reads initiated from USB by way of the PCIOUT Dedicated endpoint, the Completions are returned to the PCIIN Dedicated endpoint. For PCI Express Reads initiated by the on-chip 8051 MCU (using the **PCIMSTCTL** register), the 1-DWord Completions are always returned to the **PCIMSTDATA** register (USB Controller, offset 108h).

8.6.3.7 PCI Express Message Generation

All PCI Express Messages, except Vendor-Defined or Set_Slot_Power_Limit Messages, use the Msg Header type, which is always 4 DWords. For this type, 1 DWord of value 0000_0000h is included in the PCIOUT USB packet as Payload, and the **PCIMSTCTL** register (USB Controller, offset 100h) length is equal to 0 DWords. (Refer to Table 8-16.)

The Set_Slot_Power_Limit Message uses the MsgD Header type. A 5-DWord TLP is sent on the PCI Express Port (4-DWord Header and 1-DWord Payload). For this type, there is a 2-DWord Payload in the PCIOUT endpoint FIFO, and the **PCIMSTCTL** register length is 1 DWord. The first DWord of the USB Payload is the fourth DWord of the Message Header. The second DWord of the USB Payload is the Payload DWord of the Set_Slot_Power_Limit Message. (Refer to Table 8-17.)

Vendor-Defined Messages can be either Msg or MsgD. For Msg, there is no Payload, so this is treated the same as a standard Message with a 4-DWord PCI Express TLP. The Vendor-Defined MsgD is treated the same as the Set_Slot_Power_Limit Message, except that the Message Payload can vary from 1 to 63 DWords. The maximum Message Payload is 63 DWords, because the EP 0 OUT FIFO has space for 64 DWords, and 1 DWord is needed for the fourth DWord of the PCI Express Message Header DWord. (Refer to Table 8-18.)

Table 8-16. PCIOUT Packet Format for PCI Express Message Generation – All PCI Express Messages Except Set_Slot_Power_Limit and Vendor-Defined Messages

PCIOUT USB Packet DWord	Description
0	**PCIMSTCTL** register (USB Controller, offset 100h) contents
1	**PCIMSTADDR** register (USB Controller, offset 104h) contents
2	**PCIMSTMSG** register (USB Controller, offset 11Ch) contents

Table 8-17. PCIOUT Packet Format for PCI Express Message Generation – Set_Slot_Power_Limit Messages

PCIOUT USB Packet DWord	Description
0	**PCIMSTCTL** register (USB Controller, offset 100h) contents
1	**PCIMSTADDR** register (USB Controller, offset 104h) contents
2	**PCIMSTMSG** register (USB Controller, offset 11Ch) contents
3	Message Payload

Table 8-18. PCIOUT Packet Format for PCI Express Message Generation – Vendor-Defined Messages

PCIOUT USB Packet DWord	Description
0	**PCIMSTCTL** register (USB Controller, offset 100h) contents
1	**PCIMSTADDR** register (USB Controller, offset 104h) contents
2	**PCIMSTMSG** register (USB Controller, offset 11Ch) contents
3	Message Payload DWord 0
4	Message Payload DWord 1
…	…
65	Message Payload DWord 62

8.6.3.8 PCIOUT Flow Control

For *USB r2.0*, a NYET handshake is returned to the Host for each PCIOUT packet, indicating that the packet was accepted, but to not send any more packets. The Host pings PCIOUT. When the PCI Express transaction completes, an ACK is returned in response to a PING. The Host can then send a new packet to PCIOUT.

For *USB r3.0*, an ACK with *NumP*=0 is returned to the Host for each PCIOUT packet, indicating that the packet was accepted, but to not send any more packets. The PCIOUT endpoint is now in the Flow Control state. When the PCI Express transaction completes, an ERDY packet is sent to the Host, taking the PCIOUT endpoint out of the Flow Control state. The Host can then send a new packet to PCIOUT.

8.6.3.9 Message Broadcast

Broadcast routing for Messages is ***not supported***.

8.6.3.10 PCIOUT/PCIIN Timeout

When PCI Express Non-Posted Requests are sent to the PCI Express egress Port, a Completion Timeout timer is started. The length of time allocated to the Timer is derived from the **Device Control 2** register *Completion Timeout Value* bit (All Ports and USB Controller, offset 90h[3:0]). Programmable timeout values include:

- 00h = 20 ms
- 01h = 128 µs
- 02h = 2 ms
- 05h = 30 ms
- 06h = 200 ms
- 09h = 400 ms
- 10h = 2s
- 13h = 8s (default)
- 14h = 20s

If a Completion Timeout occurs on the PCI Express egress Port, the corresponding Port's **PCIMSTCTL** register *PCI Express Master Start* bit (USB Controller, offset 100h[6]) is Cleared. Additionally, the Port's **Uncorrectable Error Status** register *Completion Timeout Status* bit (Port 0, offset FB8h[14]) is Set. When the timeout occurs, the PCIOUT endpoint returns an ERDY to the Host, then waits for the next packet before starting another transfer. A typical Completion timeout and Recovery sequence is as follows:

1. The USB Host sends a packet to the PCIOUT endpoint, requesting a Non-Posted transaction on the PCI Express interface (*such as* a MemRd or Configuration access).
2. The USB 3382 generates the PCI Express Non-Posted Request on the PCI Express interface, then starts the Timeout Timer.
3. The PCI Express target does not respond with a Completion TLP before the Timer times out.
4. Upon timeout, the PCIOUT endpoint returns an ERDY to the USB Host. For PCI Express Read transactions, PCIIN responds to ACKs with an NRDY.
5. The Port's **Uncorrectable Error Status** register *Completion Timeout Status* bit (Port 0, offset FB8h[14]) is Set.

6. The Port's **PCIMSTCTL** register *PCI Express Master Start* bit (USB Controller, offset 100h[6]) is Cleared.

7. The USB Host detects the timeout, by polling the Port's *Completion Timeout Status* bit, by way of the CSROUT/IN endpoints.

8. The USB Host issues a new Request to PCIOUT endpoint.

8.6.3.11 PCIOUT Latency

In USB High-Speed mode, the delay from the beginning of the OUT token until the PCI Express Write transaction is complete (assuming that there are no PCI Express Target wait states) is approximately 1.5 µs. In USB Full-Speed mode, the delay is approximately 16.5 µs. Taking into account the Host latencies on a typical PC platform, the average time to perform a PCI Express Memory Write is approximately 250 µs in High-Speed mode, and 8.5 ms in Full-Speed mode.

8.6.3.12 PCIOUT STALL Conditions

OUT packets sent to the PCIOUT endpoint result in a STALL response and endpoint Halt, for the following conditions:

- USB packet size < 8 bytes
- USB packet size is not a multiple of 4 bytes
- PCI Express DWord Length (**PCIMSTCTL**[30:24] > 40h (3Fh for Messages)
- PCI Express DWord Length (**PCIMSTCTL**[30:24] > 1h and first Byte Enables are non-contiguous
- PCI Express DWord Length (**PCIMSTCTL**[30:24] > 1h and Command type is I/O or Configuration
- PCI Express DWord Length (**PCIMSTCTL**[30:24] > PCIIN FIFO size for Reads
- PCI Express DWord Length (**PCIMSTCTL**[30:24] > PCIOUT USB Maximum Packet Size - 8 (- 12 for Messages), for Writes
- USB packet size > PCI Express MPS + 8 (+ 12 for Messages)
- USB packet size > 264 (268 for Messages)
- USB packet size is not equal to (4 x PCI Express DWord Length + 8 (+ 12 for Messages)), for Writes
- PCI Express Port(s) is (are) disabled

Subsequent OUT packets to this endpoint are NAKed/NRDYed, until the PCI Express Read or Write transaction completes. Write Data Payloads are stored in the PCIOUT FIFO. For a PCI Express Write, the Reader sends an ACK signal when the transaction completes. If the selected PCI Express Link goes down, the endpoint is halted. If a malformed Completion, poisoned Completion, or Completion timeout occurs during a Non-Posted Write, a Stall handshake is returned and the endpoint is halted. The USB Controller Sets the **PCIOUT** register *Halt* bit in the *USB r2.0* or *USB r3.0* core.

8.6.4 PCIIN Endpoint

PCIIN is a Bulk endpoint that provides the PCI Express Read Completion resulting from the PCIOUT Read Request to the USB Host. Packets read from this endpoint consist of several DWords, formatted as listed in Table 8-19.

An IN token to this endpoint causes PCI Express Read data to be returned. The PCI Express Read transaction was previously initiated by a Write to the PCIOUT Dedicated endpoint. The first byte stored into the PCIIN endpoint FIFO is determined by the First Byte Enables that were written to the PCIOUT endpoint when the Read was initiated. *For example*, if the First Byte Enables value is 1100b, then Byte 2 from the PCI Express Completion is written into the PCIIN FIFO first, followed by Byte 3, and the remainder of the DWords.

If the PCI Express Read completes before the USB Data packet is provided, the endpoint ACKs. Otherwise, the endpoint NAKs/NRDYs until the Read data becomes valid. If a new OUT packet to the PCIOUT endpoint occurs after the Read data becomes valid, but before the USB Host sends the IN token to collect the data, the Read data is invalidated. An IN token that returns valid data invalidates the current Read data so that the next IN token received causes a new PCI Express Read. If the PCI Express Read transaction terminates with a Completer Abort or Unsupported Request (CA or UR, respectively), a value of FFFF_FFFFh is returned, and an optional interrupt can be generated to the CPU (8051 or PCI Express Host).

For PCIOUT-initiated Reads, FFFF_FFFFh is written to the PCIIN FIFO by the USB Controller. For 8051-initiated Reads, FFFF_FFFFh is written to the **PCIMSTDATA** register (USB Controller, offset 108h). In both cases, the **PCIIN** register *Valid* bit is Set.

For malformed Completions, bad Byte Counts, or Completion timeouts, a STALL handshake is returned and the endpoint is halted, and the USB Controller Sets the PCIIN endpoint's **DEP_RSP** register *Endpoint Halt Set* bit (USB Controller, offset 224h[8]) in the *USB r2.0* or *USB r3.0* core.

For PCI Express Configuration transactions that terminate with a Configuration Retry Status, a value of FFFF_0001h is returned to the USB Host.

For PCI Express Reads with zero length, the PCI Express Adapter (endpoint) returns a 1-DWord Payload that is placed into the PCIIN FIFO. The USB Host must either send an IN token to retrieve this DWord, or flush the PCIIN FIFO.

Table 8-19. PCIIN Endpoint Packet Format

Byte Index	Function
0 through 3	First DWord Payload (LSB first)
4 through 7	Second DWord Payload (LSB first)
…	And so forth

8.6.5 STATIN Endpoint – Root Complex Mode

The STATIN endpoint is configured as a Bulk or Interrupt endpoint that reports a change in the **IRQSTAT1** or **IRQSTAT0** status registers (USB Controller, offsets 2Ch or 28h, respectively), if the corresponding *Interrupt Enable* bit(s) are Set. When configured as an Interrupt endpoint, the maximum packet size is 8. The interrupt polling rate of this endpoint is determined by the **DEP_CFG**, **STATIN_HS_INTPOLL_RATE**, or **STATIN_FS_INTPOLL_RATE** registers (USB Controller, offset 240h, Index 84h, or Index 85h, respectively). Packets read from this endpoint are formatted as listed in Table 8-20.

For testing purposes, firmware can Set the **DIAG** register *Force USB Interrupt* bit (Index 06h[9]), which causes the STAT_IN endpoint to be validated. The *Force USB Interrupt* bit state is then reflected in the packet returned to the USB Host.

For INTERRUPT mode, if an IN token is received, and the interrupt status has not changed since the last IN token was received, then a NAK/NRDY handshake is returned to the Host. For BULK mode, the status is always returned, regardless of whether the interrupt status has changed. All interrupt sources are monitored, and if a change is detected, a *Response Enable* bit is Set. This bit enables a packet to be returned when an IN token arrives, after which, the bit is Cleared.

Table 8-20. STATIN Endpoint Packet Format

Byte Index	Description	
	Register	Bit(s)
0	**IRQSTAT1**	7:0
1		15:8
2		23:16
3		31:24
4	**IRQSTAT0**	7:0
5		13:8
	DIAG	9[a]
	Reserved (0)	15
6	***Reserved*** (0000h)	7:0
7		15:8

a. *Reflects the **DIAG** register Force USB Interrupt bit (Index 06h[9]) state.*

8.6.6 RCIN Endpoint – Root Complex Mode

RCIN is an Interrupt endpoint that enables an external PCI Express Adapter (endpoint) device to send Messages upstream, to the USB Host. The external device writes the Message to **BAR5**, which points to the RCIN endpoint FIFO. When an IN token arrives, one Message is sent to the USB Host. This endpoint defaults to a 512-byte FIFO, which can hold up to 32 Messages.

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 9 *USB r2.0* and *USB r3.0* Functional Description

9.1 Introduction

This chapter describes the following USB implementations for USB 3382:

- USB r2.0 Functional Description
- USB r3.0 Functional Description

Note: *The registers that support the USB interface are detailed in* *Chapter 16, "USB Configuration Registers."*

9.2 *USB r2.0* Functional Description

The USB 3382 is a USB-function device, and is therefore a Slave to the USB Host. The bit- and packet-level protocols, as well as the USB 3382 electrical interface, conform to the *USB r2.0*. The USB Host initiates all USB Data transfers to and from the USB 3382 USB Port. The USB 3382 is configured for up to 14 endpoints, in addition to EP 0. Six of the endpoints are dedicated, and the other eight endpoints can be of type *Isochronous*, *Bulk*, or *Interrupt*. The Configuration registers are used to program endpoint characteristics. The USB 3382 operates in Full- (12 Mbps) or High-speed (480 Mbps) mode.

Note: *The USB 3382 does not support Low-Speed mode.*

9.2.1 USB Protocol

The USB packet protocol consists of Tokens, Packets, Transactions, and Transfers. Each is described in the sections that follow.

9.2.1.1 Tokens

Tokens are a type of Packet Identifier (PID), and follow the *Sync* field at the beginning of a packet. The four classic token types are OUT, IN, SOF, and SETUP. In High-Speed mode, the USB 3382 also recognizes the PING token.

9.2.1.2 Packets

There are four types of packets – Start-of-Frame (SOF), Token, Data, and Handshake – that are transmitted and received in the order listed in Table 9-1. Each packet begins with a *Sync* field and a Packet Identifier (PID). The other fields vary, depending upon the packet type.

Table 9-1. USB Protocol Packets

Packet Type	Quantity of Bits							
	Sync Field	Packet Identifier (PID)	Frame Number	Address	Endpoint	Data	Cyclic Redundancy Checks (CRC)	Total
1. Start-of-Frame (SOF)	8	8	11	–	–	–	5	32
2. Token	8	8	–	7	4	–	5	32
3. Data	8	8	–	–	–	*N*	16	32 + *N*
4. Handshake	8	8	–	–	–	–	–	16

9.2.1.3 Transactions

A single transaction consists of a Token packet, optional Data packet(s), and a Handshake packet.

9.2.1.4 Transfers

A single transfer consists of one or more transactions. Control transfers consist of a Setup transaction, optional Data transactions, and a Status transaction.

9.2.2 Automatic Retries

9.2.2.1 OUT Transactions

If an error occurs during an OUT transaction, the USB 3382 re-loads its USB FIFO Write pointer to the beginning of the failed packet. The Host then transmits another OUT token and re-transmits the packet. After the USB 3382 successfully receives the packet, the endpoint's **EP_STAT** register *Data Packet Received Interrupt* bit (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[3]) is Set. The USB 3382 can handle an unlimited quantity of back-to-back Retries; however, the Host determines the quantity of packet Retries.

9.2.2.2 IN Transactions

If an error occurs during an IN transaction, the USB 3382 re-loads its USB FIFO Read pointer to the beginning of the failed packet. The Host then transmits another IN token, and the USB 3382 re-transmits the packet. After the Host successfully receives the packet, the endpoint's **EP_STAT** register *Data Packet Transmitted Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[2]) is Set.

9.2.3 PING Flow Control

When operating in High-Speed mode, the USB 3382 supports the PING protocol for Bulk OUT and Control endpoints. This protocol allows the USB 3382 to indicate to the Host that it cannot accept an OUT packet. The Host then transmits PING tokens, to query the USB 3382. The USB 3382 returns an ACK in response to the PING when it is able to accept a maximum-size packet. At this time, the Host transmits an OUT token and Data packet. The USB 3382 returns an ACK handshake if the packet is accepted, and there is sufficient space to receive an additional packet. The USB 3382 returns a NYET handshake to the Host if it can accept only the current packet. The Host then starts transmitting PING tokens.

9.2.4 Packet Sizes

An endpoint Maximum Packet Size is determined by the corresponding **EP_*n*_MAXPKT** register. For IN transactions, the USB 3382 returns a maximum-size packet to the Host if the quantity of “Maximum Packet” bytes exist in the FIFO. If the FIFO data is validated, a packet size less than the maximum is returned to the Host in response to an IN token. For Interrupt and Isochronous endpoints, the Maximum Packet Size must be an even multiple of 8. Table 9-2 lists the allowable Maximum Packet Sizes, by endpoint type.

Table 9-2. Allowable Maximum Packet Sizes, By Endpoint Type

Endpoint Type	Allowable Maximum Packet Size (Bytes)	
	Full-Speed Mode	High-Speed Mode
Bulk	8, 16, 32, 64	512
Control	8, 16, 32, 64	64
Interrupt	64 maximum	1,024 maximum
Isochronous	1,023 maximum	1,024 maximum

9.2.5 USB Endpoints

The USB 3382 supports Bulk, Control, Interrupt, and Isochronous endpoints. All endpoints, except for Control endpoints, are unidirectional. Bidirectional Bulk, Interrupt, and Isochronous traffic requires two endpoints.

9.2.5.1 Control Endpoint (EP 0)

The Control endpoint, EP 0 (EP 0), is a ***reserved*** endpoint. The USB Host uses this endpoint to configure and acquire information about the USB 3382, its configurations, interfaces, and other endpoints. Control endpoints are bidirectional, and data delivery is guaranteed.

The Host transmits 8-byte Setup packets to EP 0, to which the USB 3382 interprets and responds. The USB 3382 consists of a set of registers dedicated to storing the Setup packet, and uses the EP 0 FIFO for Control data. For Control Writes, data flows through the FIFO from the USB interface to the PCI Express interface. For Control Reads, data flows through the FIFO from the PCI Express interface to the USB interface.

When EP 0 detects a Setup packet, the USB 3382 Sets *Status* bits in the endpoint's **EP_STAT** register (USB Controller, offset 30Ch) and interrupts the CPU (8051 or PCI Express Host). The CPU reads the Setup packet from the **EP_STAT** register, and responds based upon the packet contents. The CPU provides data to return to the Host, including status and Descriptors, unless the corresponding **STDRSP** register (USB Controller, offset 80h) auto-enumerate bit is Set. Refer to the *USB r2.0*, Chapter 9, "USB Standard Device Requests," for a description of the data that must be returned for each USB Request. The Host rejects Descriptors that contain unexpected field values.

Control Write Transfer

A successful Control Write transfer to Control EP 0 consists of the data listed in Table 9-3.

During the Setup transaction, the USB 3382 stores the Data Stage packet in its Setup registers. The USB 3382 returns an ACK handshake to the Host after all eight bytes are received. A *Setup Packet Interrupt* bit is Set to notify the CPU (8051 or PCI Express Host) that a Setup packet was received. The CPU reads and interprets the 8-byte Data packet. A Setup transaction cannot be stalled or NAKed; however, if the data is corrupt, the USB 3382 does not return an ACK to the Host.

During the optional Data transaction, zero, one, or more Data packets are written into the EP 0 FIFO. For each packet:

1. Interrupt bits are Set and can interrupt the CPU.
2. CPU reads the FIFO.
3. USB 3382 returns an ACK if no error occurred.

For a successful Status transaction, the USB 3382 returns a zero-length Data packet. A NAK or STALL handshake is returned if an error occurs.

Table 9-3. Control Write Transfer

Stage	Packet	Packet Contents	Quantity of Bytes	Source
Setup	Setup Token	SETUP PID, address, endpoint, and CRC5	3	Host
	Data	DATA0 PID, 8 data bytes, and CRC16	11	Host
	Status	ACK	1	USB 3382
Data (zero, one, or more packets)	OUT Token	OUT PID, address, endpoint, and CRC5	3	Host
	Data (1/0)	DATA PID, *N* data bytes, and CRC16	*N*+3	Host
	Status	ACK	1	USB 3382
Status	IN Token	IN PID, address, endpoint, and CRC5	3	Host
	Data	DATA1 PID, zero-length packet (ZLP), and CRC16	3	USB 3382
	Status	ACK	1	Host

Control Write Transfer Details

For Control Write transfers, the Host first transmits 8 bytes of Setup information. The Setup bytes are stored into an 8-byte register bank, accessed by the CPU (8051 or PCI Express Host). After the 8 bytes are stored into the Setup registers, the *Setup Packet Interrupt* bit is Set. The CPU then reads the 8-byte Setup packet and prepares to respond to the optional Data stage. The quantity of bytes to be transferred in the Data stage is specified in the Setup packet. When the Setup packet is received, the *Control Status Stage Handshake* bit is automatically Set, in anticipation of the Control Status stage. While this bit is Set, the Control Status stage is acknowledged with a NAK, allowing the CPU to prepare its handshake response (ACK or STALL). After the *Control Status Stage Handshake* bit is Cleared and the OUT FIFO is empty, an ACK or STALL handshake is returned to the Host. Waiting for the OUT FIFO to become empty prevents another Control Write from corrupting the current packet data in the FIFO.

During a Control Write operation, an optional Data stage can follow the Setup stage. The *Data OUT Token Interrupt* bit is Set at the beginning of each Data packet. The bytes corresponding to the Data packet are stored into the EP 0 FIFO. If the FIFO fills and the Host transfers another byte, the USB 3382 returns a NAK handshake to the Host, signaling that the data cannot be accepted.

If a packet is not successfully received (NAK or Timeout status), the endpoint's **EP_STAT** register *Data Packet Received Interrupt* bit (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[3]) is not Set, and the data is automatically flushed from the FIFO. The Host later re-transmits the same packet. This process is transparent to the CPU.

If the CPU halted this endpoint by Setting the *Endpoint Halt* bit, the USB 3382 does not store data into the FIFO, and responds with a STALL acknowledge to the Host. There is no Status stage in this case.

The CPU can poll the *Data Packet Received Interrupt* bit or enable the bit as an interrupt, then read the packet from the FIFO. If the Host tries to write more data than indicated in the Setup packet, the CPU Sets EP 0's *Endpoint Halt* bit. In this case, there is no Status stage from the Host.

After all optional Data Stage packets are received, the Host transmits an IN token, signifying the Status stage. The *Control Status Interrupt* bit is Set after the Status stage IN token is received. Until the CPU Clears the *Control Status Stage Handshake* bit and the OUT FIFO is empty, the USB 3382 responds with NAKs, indicating that the USB 3382 is processing the Setup command. When the CPU Clears the *Control Status Stage Handshake* bit and firmware has emptied the OUT FIFO data, the USB 3382 responds with a Zero-Length Data packet (transfer OK) or STALL (error encountered).

Control Read Transfer

A successful Control Read transfer from Control EP 0 consists of the data listed in Table 9-4.

The Setup transaction is processed with the same method as Control Write transfers. (Refer to "Control Write Transfer Details.") During the optional Data transaction, zero, one, or more Data packets are read from the EP 0 FIFO. For each packet:

1. Interrupt bits are Set and can interrupt the CPU (8051 or PCI Express Host).
2. CPU writes data to the FIFO.
3. If there is no data in the FIFO, a NAK or ZLP is returned to the Host.
4. Host returns an ACK to the USB 3382 if no error occurred.

For a successful Status stage, the Host transmits a Zero-Length Data packet, and the USB 3382 responds with an ACK. A NAK or STALL is returned if an error occurred.

Table 9-4. Control Read Transfer

Stage	Packet	Packet Contents	Quantity of Bytes	Source
Setup	Setup Token	SETUP PID, address, endpoint, and CRC5	3	Host
	Data	DATA0 PID, 8 data bytes, and CRC16	11	Host
	Status	ACK	1	USB 3382
Data (zero, one, or more packets)	IN Token	IN PID, address, endpoint, and CRC5	3	Host
	Data (1/0)	DATA PID, *N* data bytes, and CRC16	*N*+3	USB 3382
	Status	ACK	1	Host
Status	OUT Token	OUT PID, address, endpoint, and CRC5	3	Host
	Data	DATA1 PID, ZLP, and CRC5	3	Host
	Status	ACK	1	USB 3382

Control Read Transfer Details

For Control Read transfers, the Host first transmits eight bytes of Setup information. The Setup bytes are stored into an 8-Byte Register bank that is accessed from the CPU (8051 or PCI Express Host). After the 8 bytes are stored in the Setup registers, the *Setup Packet Interrupt* bit is Set. The CPU then reads the 8-Byte Setup packet and prepares to respond to the optional Data stage. The quantity of bytes to be transferred in the Data stage is specified in the Setup packet. When the Setup packet is received, the *Control Status Stage Handshake* bit is automatically Set. While this bit is Set, the Control Status stage is acknowledged with a NAK, allowing the CPU to prepare its handshake response (ACK or STALL). After the *Control Status Stage Handshake* bit is Cleared, an ACK or STALL handshake is returned to the Host.

During a Control Read operation, an optional Data stage can follow the Setup stage. After the Setup stage, the CPU can start writing the first byte of packet data into the EP 0 FIFO, in anticipation of the Data stage. The *Data In Token Interrupt* bit is Set at the beginning of each Data packet. If there is data in the EP 0 FIFO, the data is returned to the Host. If EP 0 has no data to return, the endpoint returns a ZLP (signaling that no further data is available) or NAK handshake (the data is not available yet, but will be soon). The USB 3382 responds to the Data stage IN tokens, according to the data listed in Table 9-5.

After each packet is transmitted to the Host, the EP 0 **EP_STAT** register *Data Packet Transmitted Interrupt* bit (USB Controller, offset 30Ch[2]) is Set.

If a packet is not successfully transmitted (*Timeout* status bit Set), the *Data Packet Transmitted Interrupt* bit is not Set, and this packet is transmitted to the Host when another IN token is received. The Retry operation is transparent to the CPU.

If the Host tries to read more data than requested in the Setup packet, the CPU Sets the *Endpoint Halt* bit for the endpoint.

After all optional Data Stage packets are transmitted, the Host transmits an OUT token, followed by a Zero-Length Data packet, signifying the Status stage. The *Control Status Interrupt* bit is Set after the Status stage OUT token is received. Until the CPU Clears the *Control Status Stage Handshake* bit, the USB 3382 responds with NAKs, indicating that the USB 3382 is processing the command specified by the Setup stage. When the CPU Clears the *Control Status Stage Handshake* bit, the USB 3382 responds with an ACK (transfer OK) or STALL (EP 0 is stalled).

Table 9-5. Control Read Transfer Details

Packet Validated	Amount of Data in FIFO	Action
0	< Maximum Packet Size	NAK to the Host
X	≥ Maximum Packet Size	Return data to the Host
1	Empty	Zero-Length packet to the Host
1	> 0	Return data to the Host

Notes: *“X” is “Don’t Care.”*

9.2.5.2 Isochronous Endpoints

Isochronous endpoints are used for time-critical Data transfers. Isochronous transfers do not support handshaking nor error-checking protocol, and are guaranteed a certain amount of bandwidth during each frame. The USB 3382 ignores Cyclic Redundancy Checks (CRC) and Bit-Stuffing errors during Isochronous transfers; however, the USB 3382 Sets the endpoint's **EP_STAT** register (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch) handshaking status bits the same as it Sets for Non-Isochronous packets, enabling the CPU (8051 or PCI Express Host) to detect the errors. Isochronous endpoints are unidirectional, with the direction defined by the **EP_CFG** register(s) (USB Controller, offset(s) 300h, 320h, 340h, 360h, 380h).

For Isochronous endpoints, the Packet FIFO Size must be equal to or greater than the Maximum Packet Size. The Maximum Packet Size for an Isochronous endpoint ranges from 1 to 1,024 bytes, and must be an even multiple of 8 bytes.

For an Isochronous OUT endpoint, the CPU or DMA Controller can read data from the FIFO after an entire packet is received. If the FIFO is the same size as the Maximum Packet Size, then the ISO bandwidth must be Set so that the FIFO is emptied before the next ISO packet arrives.

For an Isochronous IN endpoint, the CPU or DMA Controller can write data to the FIFO at the same time that data is being transmitted to the USB.

Isochronous OUT Transactions

Isochronous OUT endpoints transfer data from a USB Host to the PCI Express interface. An Isochronous OUT transaction consists of the data listed in Table 9-6.

The USB Host initiates an Isochronous OUT transaction by transmitting an OUT token to an Isochronous OUT endpoint. The *Data OUT Token Interrupt* bit is Set when the OUT token is recognized. The bytes corresponding to the Data packet are stored into the endpoint's FIFO. Isochronous transactions are not Retried; therefore, if the FIFO is full when the Host transfers a packet (or the *NAK Packets* bit is Set), the packet is discarded. No Handshake packets are returned to the Host; however, the *USB OUT ACK Transmitted* and *Timeout* status bits are Set to indicate transaction status. If a CRC error is detected, the packet is accepted, and the *Timeout* status bit is Set. After all Data packets are received, the CPU (8051 or PCI Express Host) samples these status bits to determine whether the USB 3382 successfully received the packet.

By definition, Isochronous endpoints do not use handshaking with the Host. Because there is no method to return a stalled handshake from an Isochronous endpoint to the Host, data that is transmitted to a stalled Isochronous endpoint is received normally. The Maximum Packet Size must be less than or equal to the FIFO size.

If small packets are transmitted to this endpoint, the PCI Express interface must read a packet from the FIFO before the Host transmits the next packet. *For example*, if the Host transmits two 1-byte packets, each byte occupies one line in the FIFO. The endpoint's **EP_AVAIL** register(s) (USB Controller, offset(s) 310h, 330h/3F0h, 350h/410h, 370h/430h, 390h/450h[13:0]) reports that two bytes are available; however, there is no indication that two 32-bit Reads are required to access the bytes. Only one byte is valid for each 32-bit Read; the other bytes are ignored.

The CPU or DMA Controller wait for the endpoint's **EP_STAT** register *Data Packet Received Interrupt* bit (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[3]) to be Set. After the interrupt is Set, the data is read from the FIFO.

Table 9-6. Isochronous OUT Transactions

Packet	Packet Contents	Quantity of Bytes	Source
OUT Token	OUT PID, address, endpoint, and CRC5	3	Host
Data	DATA0 PID, *N* data bytes, and CRC16	*N*+3	Host

High-Bandwidth Isochronous OUT Transactions

The Host transmits high-bandwidth OUT PID sequences for each microframe, depending upon the Endpoint Descriptor **GPEP[3:0/Out/In]_HS_MAXPKT** register *Additional Transaction Opportunities* field(s) (USB Controller, Index 20h, 30h, 40h, 50h, 60h, 70h, 80h, 90h[12:11]), as listed in Table 9-7.

The USB 3382 accepts data (unless the FIFO is full), and records the PID in the endpoint's **EP_STAT** register *High-Bandwidth OUT Transaction PID* field(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[23:22]). This allows firmware to track PIDs as they arrive and determine whether the data sequence is complete. (Refer to Table 9-8.)

Table 9-7. High-Bandwidth Isochronous OUT Transactions

Additional Transaction Opportunities Field	PID Sequence
00b	DATA0 (standard ISO)
01b	MDATA, DATA1 (one additional transaction)
10b	MDATA, MDATA, DATA2 (two additional transactions)

Table 9-8. High-Bandwidth Isochronous OUT Transactions Data Sequence

High-Bandwidth OUT Transaction PID Field	PID Received
00b	DATA0
01b	DATA1
10b	DATA2
11b	MDATA

Isochronous IN Transactions

Isochronous IN endpoints transfer data from the PCI Express interface to a USB Host. An Isochronous IN transaction consists of the data listed in Table 9-9.

The USB Host initiates an Isochronous IN transaction by transmitting an IN token to an Isochronous IN endpoint. The *Data IN Token Interrupt* bit is Set when the IN token is recognized. If there is data in the endpoint's FIFO, the data is returned to the Host. If the endpoint has no data to return, a Zero-Length packet is returned to the Host. The USB 3382 responds to the IN token, according to the data listed in Table 9-10.

After the packet is transmitted to the Host, the endpoint's **EP_STAT** register *Data Packet Transmitted Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[2]) is Set. If an IN token arrives and there is no valid packet in the FIFO, the USB 3382 returns a Zero-Length packet. No Handshake packets are returned to the Host; however, the *USB IN ACK Transmitted* and *Timeout* status bits are Set, to indicate transaction status. After all Data packets are transmitted, the CPU (8051 or PCI Express Host) samples these status bits, to determine whether the packets were successfully transmitted to the Host.

By definition, Isochronous endpoints do not handshake with the Host. Because there is no method to return a stalled handshake from an Isochronous endpoint to the Host, data that is requested from a stalled Isochronous endpoint is transmitted normally.

Table 9-9. Isochronous IN Transactions

Packet	Packet Contents	Quantity of Bytes	Source
IN Token	IN PID, address, endpoint, and CRC5	3	Host
Data	DATA0 PID, *N* data bytes, and CRC16	*N*+3	USB 3382

Table 9-10. IN Token Response

Packet Validated	Amount of Data in FIFO	Action
0	< Maximum Packet Size	Zero-Length packet to the Host; *USB IN NAK Transmitted* status bit Set
X	≥ Maximum Packet Size	Return data to the Host
1	Empty	Zero-Length packet to the Host
1	> 0	Return data to the Host

Note: *"X" is "Don't Care."*

High-Bandwidth Isochronous IN Transactions

A USB device that provides High-Bandwidth ISO IN endpoints is required to transmit ISO PID sequences for each microframe, according to the Endpoint Descriptor **GPEP[3:0/Out/In]_HS_MAXPKT** register *Additional Transaction Opportunities* field(s) (USB Controller, Index 20h, 30h, 40h, 50h, 60h, 70h, 80h, 90h[12:11]), as listed in Table 9-11.

When the first microframe IN token arrives, the USB 3382 copies the *Additional Transaction Opportunities* field, to determine the initial PID. Upon each succeeding microframe IN token, the PID advances to the next token.

Table 9-11. High-Bandwidth Isochronous IN Transactions

Additional Transaction Opportunities Field	PID Sequence
00b	DATA0 (standard ISO)
01b	DATA1, DATA0 (one additional transaction)
10b	DATA2, DATA1, DATA0 (two additional transactions)

9.2.5.3 Bulk Endpoints

Bulk endpoints are used for guaranteed error-free delivery of large amounts of data between a Host and device. Bulk endpoints are unidirectional, with the direction defined by the **EP_CFG** register(s) (USB Controller, offset(s) 300h, 320h, 340h, 360h, 380h).

Bulk OUT Transactions

Bulk OUT endpoints transfer data from a USB Host to the PCI Express interface. A Bulk OUT transaction to a Bulk OUT endpoint consists of the data listed in Table 9-12.

The USB Host initiates a Bulk OUT transaction by transmitting an OUT token to a Bulk OUT endpoint. The *Data OUT Token Interrupt* bit is Set when the OUT token is recognized. The bytes corresponding to the Data packet are stored into the endpoint's FIFO. If the FIFO is full when the Host transfers another packet, the packet is discarded and the *USB OUT NAK Transmitted* status bit is Set. At packet completion, a NAK handshake is returned to the Host, indicating that the packet cannot be accepted.

All USB data passes through the endpoint's FIFO to the PCI Express interface. The CPU waits until the Data Packet Received interrupt occurs before reading the FIFO data.

If a packet is not successfully received (*USB OUT NAK Transmitted* or *Timeout* status bits Set), the *Data Packet Received Interrupt* bit is not Set, and the data is automatically flushed from the FIFO. The Host later retransmits the same packet. This process is transparent to the CPU (8051 or PCI Express Host).

If the CPU halted this endpoint by Setting the *Endpoint Halt* bit, the USB 3382 does not store data into the FIFO, and responds with a STALL handshake to the Host.

Table 9-12. Bulk OUT Transactions

Packet	Packet Contents	Quantity of Bytes	Source
OUT Token	OUT PID, address, endpoint, and CRC5	3	Host
Data (1/0)	DATA PID, *N* data bytes, and CRC16	*N*+3	Host
Status	ACK, NAK, or STALL	1	USB 3382

Bulk IN Endpoints

Bulk IN endpoints transfer data from the PCI Express interface to a USB Host. A Bulk IN transaction from a Bulk IN endpoint consists of the data listed in Table 9-13.

The USB Host initiates a Bulk IN transaction by transmitting an IN token to a Bulk IN endpoint. The *Data IN Token Interrupt* bit is Set when the IN token is recognized. If there is validated data in the endpoint's FIFO, the data is returned to the Host. If the endpoint has no data to return, a Zero-Length packet (signaling that no further data is available) or NAK handshake (the data is not available yet) is returned. The USB 3382 responds to the IN token, according to the data listed in Table 9-14.

After the packet is transmitted to the Host, the endpoint's **EP_STAT** register *Data Packet Transmitted Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[2]) is Set.

If a packet is not successfully transmitted (*Timeout* status bit is Set), the *Data Packet Transmitted Interrupt* bit is not Set, and the same packet is transmitted to the Host when another IN token is received. The Retry operation is transparent to the CPU (8051 or PCI Express Host).

If the CPU halted this endpoint by Setting the *Endpoint Halt* bit, the USB 3382 responds to the IN token with a STALL handshake to the Host.

Table 9-13. Bulk IN Endpoints

Packet	Packet Contents	Quantity of Bytes	Source
IN Token	IN PID, address, endpoint, and CRC5	3	Host
Data (1/0)	DATA PID, *N* data bytes, and CRC16, or NAK or STALL	*N*+3	USB 3382
Status	ACK	1	Host

Table 9-14. Bulk IN Endpoints Packet Validation

Packet Validated	Amount of Data in FIFO	Action
0	< Maximum Packet Size	NAK to the Host
X	≥ Maximum Packet Size	Return data to the Host
1	Empty	Zero-Length packet to the Host
1	> 0	Return data to the Host

Note: *"X" is "Don't Care."*

9.2.5.4 Interrupt Endpoints

Interrupt endpoints are used for transmitting or receiving small amounts of data to the Host with a bounded service period.

Interrupt OUT Transactions

Interrupt OUT endpoints transfer data from a USB Host to the PCI Express interface. An Interrupt OUT transaction to an Interrupt OUT endpoint consists of the data listed in Table 9-15.

The behavior of an Interrupt OUT endpoint is essentially the same as a Bulk OUT endpoint, except for the *Endpoint Toggle* bit. If the **EP_RSP** register *Interrupt Mode* bit is Cleared, the Interrupt OUT *Endpoint Toggle* bit is initialized to 0 (DATA0 PID), and behaves the same as a Bulk OUT endpoint. If the *Interrupt Mode* bit is Set, the Interrupt OUT *Endpoint Toggle* bit changes after each Data packet is received from the Host, without regard to the Status packet. PING protocol is not allowed for Interrupt OUT endpoints, as per the *USB r2.0*.

Table 9-15. Interrupt OUT Transactions

Packet	Packet Contents	Quantity of Bytes	Source
OUT Token	OUT PID, address, endpoint, and CRC5	3	Host
Data (1/0)	DATA PID, *N* data bytes, and CRC16	*N*+3	Host
Status	ACK, NAK, or STALL	1	USB 3382

Interrupt IN Endpoints

An Interrupt IN endpoint is polled at a rate specified in the Endpoint Descriptor. An Interrupt transaction from an Interrupt IN endpoint consists of the data listed in Table 9-16.

The behavior of an Interrupt IN endpoint is the same as a Bulk IN endpoint, except for the toggle bit. If the *Interrupt Mode* bit is Cleared, the Interrupt IN *Endpoint Toggle* bit is initialized to 0 (DATA0 PID), and behaves the same as a Bulk IN endpoint. Use Interrupt endpoints to communicate rate feedback information for certain Isochronous functions. To support this mode, the *Interrupt Mode* bit is Set, and the Interrupt IN *Endpoint Toggle* bit changes after each Data packet is transmitted to the Host, without regard to the Status packet.

After the packet is validated and transmitted to the Host, the USB 3382 does ***not*** return a Zero-Length packet in response to the next IN token.

Table 9-16. Interrupt IN Endpoints

Packet	Packet Contents	Quantity of Bytes	Source
IN Token	IN PID, address, endpoint, and CRC5	3	Host
Data (1/0)	DATA PID, *N* data bytes, and CRC16	*N*+3	USB 3382
Status	ACK	1	Host

High-Bandwidth Interrupt Endpoints

From the USB device point of view, high-bandwidth Interrupt endpoints are the same as Bulk endpoints, except that the MAXPKT is a value from 8 to 1,024, in even multiples of 8. Standard Interrupt endpoints in Full-Speed mode can program MAXPKT to a value from 8 to 64, in even multiples of 8.

9.2.6 USB Test Modes

Use the **XCVRDIAG** register *Force High-Speed Mode* or *Force Full-Speed Mode* bits (USB Controller, offset 94h[31:30]) to force the USB 3382 into High- or Full-Speed mode, respectively. These bits ***must not be used in standard operation***, because they are for testing purposes only. In standard operation, the USB 3382 automatically performs *USB r2.0*-Chirp Protocol negotiation with the Host, to determine the correct operating speed.

USB r2.0 Test mode support is provided by way of the register's *USB Test Mode* field (field [26:24]). This field selects the appropriate USB Test mode Settings. (Refer to the *USB r2.0,* Section 9.4.9, for further details.) Usually, the Host transmits a SET_FEATURE request with the Test Selector in the upper byte of `wIndex`. To select the correct Test mode, copy the Test Selector into the *USB Test Mode* field.

USB Test mode Settings are functional only when the USB 3382 is in High-Speed mode. Also, if the USB 3382 is operating in High-Speed mode, and the *USB Test Mode* field is programmed to a non-zero value, the USB 3382 is prevented from switching out of High-Speed mode. Standard USB Suspend and Reset, as well as the *Force High-Speed Mode* and *Force Full-Speed Mode* bits, are ignored for testing purposes.

The USB 3382 can be forced into High-Speed mode (using the *Force High-Speed Mode* bit), regardless of whether the USB 3382 is connected to a Host Controller. After choosing High-Speed mode, USB Test modes can be selected.

Most USB Test modes require no further support from the USB 3382 firmware. However, the Test_Packet (100b) Test Selector requires the USB 3382 to return a specific packet.

Firmware performs the following sequence to activate the test mode:

1. Programs the *USB Test Mode* field to 100b (Test Packet).
2. Flushes EP 0.
3. Loads the following 53 (35h) hex byte packets into EP 0:

   00 00 00 00 00 00 00 00 - 00 AA AA AA AA AA AA AA AA EE EE EE EE EE EE -

   EE EE FE FF FF FF FF FF FF FF FF FF FF FF 7F BF DF - EF F7 FB FD FC 7E BF DF EF

   F7 FB FD 7E

The packet is validated, using the EP 0 **EP_CFG** register *EP FIFO Byte Count* field (USB Controller, offset 300h[18:16]).

Test modes can be auto-responded, in which case the Test Packet (100b) Test Mode automatically loads the specified Test packet. Test modes can also be manually loaded, in which case Test packets (up to 64 bytes) can be loaded into the EP 0 FIFO.

9.3 *USB r3.0* Functional Description

9.3.1 USB Interface

The USB 3382 is a USB functional device, meaning that its USB interface connects only to a USB Host or the downstream Port of a USB hub. The bit- and packet-level protocols, as well as the electrical interface, conform to the *USB r3.0*. The USB Host initiates all USB Data transfers to and from Port 0. The USB 3382 is configured for up to 14 endpoints, in addition to the Control endpoint, EP 0. Six of the endpoints are dedicated, and the other eight can be of type Isochronous, Bulk, or Interrupt. Configuration registers are used to program endpoint characteristics. The USB 3382 operates in Full-Speed (12 Mbps), High-Speed (480 Mbps), or SuperSpeed (5 Gbps) mode.

Full- and High-Speed information is discussed in Section 9.2. Because SuperSpeed mode is new (to *USB r3.0*), only SuperSpeed mode-related information is provided in this section.

Note: *The USB 3382 does not support Low-Speed mode.*

9.3.2 USB Protocol

The SuperSpeed USB packet protocol consists of four SuperSpeed packet types:

- Link Management Packet (LMP)
- Transaction Packet (TP)
- Data Packet (DP)
- Isochronous Timestamp Packet (ITP)

All packets start with a 14-byte Header, followed by a 2-byte Link Control word at the end of the packet Header (16 bytes total).

Note: *The USB 3382 does not support the Stream protocol.*

9.3.2.1 Link Management Packet

Link Management Packets (LMPs) are packets that are exchanged only between the SuperSpeed USB device and the hub or Host to which it is directly connected. LMPs are used to manage the SuperSpeed Link. LMPs are identified as *Type* field, with a value of 00000b in the packet format. The USB 3382 supports the following LMP subtypes, as defined in the *USB r3.0*, Section 8.4.1:

- Set Link Function
- U2 Inactivity timeout
- Port Capability
- Port Configuration
- Port Configuration Response

Note: *The LMP subtypes listed above are a subset of the LMP types defined in the USB r3.0, Table 8-3.*

9.3.2.2 Transaction Packet

Transaction packets (TPs) are used to control data flow and manage the end-to-end connection. TPs have no Data Payload. Each TP has a 16-bit Link Control word and 16-bit Cyclic Redundancy Check (CRC) to protect the data integrity. The USB 3382 supports the following TP subtypes:

- ACK
- NRDY
- ERDY
- STATUS
- STALL
- DEV_NOTIFICATION
- PING
- PING RESPONSE

9.3.2.3 Data Packet

Data packets (DPs) can be sent by either the USB Host or device. The USB 3382 uses DPs to return data to the Host in response to an ACK TP. All Data packets are comprised of a Data Packet Header (DPH) and Data Packet Payload (DPP). A 32-bit CRC-32 check sum follows the DPP.

Note: *Refer to the USB r3.0, Section 7.2.1.2, for details regarding the Data Packet structure.*

9.3.2.4 Isochronous Timestamp Packet

An Isochronous Timestamp Packet (ITP) is a Multicast packet sent by the Host, to all active Links. It consists of the following two fields:

- *Isochronous Timestamp* (ITS) provides a 14-bit Bus Interval Counter and 13-bit delta from the start of the current ITP to the previous bus interval boundary
- *Bus Interval Adjustment Control*

9.3.3 Transactions and Transfer

Control transfers consist of a Setup transaction stage, optional Data transaction stage, and Status transaction stage.

Bulk transactions use a two-phase transaction consisting of TPs and DPs. Under certain Flow Control and Halt conditions, the Data transaction stage can be replaced with a TP. The USB 3382 supports Burst transfers for Bulk transactions.

The Interrupt Transfer type is used for infrequent Data transfers with a bounded service period, which is similar to Bulk transactions. Interrupt transfers are limited to a burst of three DPs within each service interval. The Host is required to send an ACK TP for every DP received within the service interval, even if it is the last DP in that service interval. The final ACK TP acknowledges the last DP received, and issues a command to Clear the *NumP* field.

9.3.4 Automatic Retries

9.3.4.1 OUT Transactions

If an error occurs during an OUT transaction, the USB 3382 re-loads its USB FIFO Write pointer to the beginning of the failed packet, then returns an ACK (Retry) TP to the Host. The Host then re-transmits the packet. After the USB 3382 successfully receives the packet, the endpoint's **EP_STAT** register *Data Packet Received Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[3]) is Set. The USB 3382 can handle an unlimited quantity of back-to-back Retries; however, the Host determines the quantity of packet Retries.

9.3.4.2 IN Transactions

If an error occurs during an IN transaction, and the Host returns an ACK for the IN Data packet received with the *Retry* bit Set, the USB 3382 responds with an NRDY, followed by an ERDY. Meanwhile, the USB 3382 rewinds its USB FIFO Read pointer to the last ACKed Data packet received from the Host. The Host then transmits another ACK TP, and the USB 3382 re-transmits the packet. After the Host successfully receives the packet, the endpoint's **EP_STAT** register *Data Packet Transmitted Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[2]) is Set.

9.3.5 *USB r3.0* Flow Control

SuperSpeed USB has its own Flow Control mechanism, different from those of Full- and High-Speed. Only Bulk, Control, and Interrupt endpoints can send Flow Control responses.

An IN endpoint is considered to be in a Flow Control condition, if it returns the following responses to an ACK TP:

- Responding with an NRDY TP, –or–
- Sending a DP with the *EOB* field Set in the DPH

An OUT endpoint is considered to be in the Flow Control condition, if it returns one of the following responses to a DP:

- Responding with an NRDY TP, –or–
- Sending an ACK TP with the *NumP* field Cleared

When an endpoint is in a Flow Control condition, it sends an ERDY TP to be moved back into the *Active* state. If the endpoint is an IN endpoint, it waits until it receives an ACK TP for the last DP it transmitted, before it can send an ERDY TP.

9.3.6 Packet Sizes

An endpoint Maximum Packet Size is determined by the corresponding **EP_*n*_MAXPKT** register. For IN transactions, the USB 3382 returns a maximum-sized packet to the Host, if the quantity of "Maximum Packet" bytes exist in the FIFO. If the FIFO data is validated, a packet size less than the maximum size can be returned to the Host. Table 9-17 lists the allowable Maximum Packet Sizes. Interrupt and Isochronous endpoints must have a Maximum Packet Size that is an even multiple of 8.

Table 9-17. Allowable Maximum Packet Size

Endpoint Type	Allowable Maximum Packet Size (Bytes), by Mode		
	Full-Speed	High-Speed	SuperSpeed
Control	8, 16, 32, 64	64	512
Bulk	8, 16, 32, 64	512	1,024
Interrupt	64 maximum	1,024 maximum	1,024 maximum
Isochronous	1,023 maximum	1,024 maximum	1,024 maximum

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 10 DMA Controller

10.1 Overview

The USB 3382 consists of four DMA channels, which are used to transfer data between the USB endpoint FIFOs and PCI Express interface. The four channels are assigned to General-Purpose Endpoints 0, 1, 2, and 3 (GPEP[3:0]). Each channel performs a single or Scatter/Gather DMA Data transfer, by processing a linked list of Descriptors. The DMA starting address can be on any byte boundary.

Note: *Registers that support the DMA Controller are detailed in Section 16.8, "DMA Registers."*

10.2 Single Transfer Mode

In Single Transfer mode, a single block of data is transferred between a USB FIFO and the PCI Express interface.

10.2.1 OUT Endpoints

When a packet is received into an OUT endpoint that has an assigned DMA channel, a Write Request is made to the PCI Express space. The DMA Controller then reads data from the endpoint FIFO, and writes the data to PCI Express space. These transactions continue until the DMA Byte Count reaches 0. If the FIFO becomes empty, the DMA Controller pauses until additional data is available. The USB Controller registers listed in Table 10-1 must be programmed for a single DMA Read transfer.

Writing to the channel's **DMASTAT** register *DMA Start* bit(s) (USB Controller, offset(s) 184h, 1A4h, 1C4h, 1E4h[0]) starts the DMA Data transfer. The DMA Data transfer can also be automatically started when an OUT packet is received, if the channel's **DMACTL** register *DMA OUT Auto Start Enable* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[4]) is Set. When the DMA Data transfer is complete, various interrupts are generated.

Table 10-1. OUT Endpoints – Registers to Program for Single DMA Read Transfers

Offset(s)	Register	Description	Notes
180h, 1A0h, 1C0h, 1E0h	**DMACTL**	DMA Control	Select Single Transfer mode and other miscellaneous controls
190h, 1B0h, 1D0h, 1F0h	**DMACOUNT**	DMA Transfer Length	Set to the quantity of bytes and the direction to transfer
194h, 1B4h, 1D4h, 1F4h	**DMAADDR**	DMA Address	Set to the PCI Target address

10.2.2 IN Endpoints

When an IN endpoint is ready to transmit a packet to the Host and there is sufficient space available in the endpoint FIFO, a Read Request is issued to the PCI Express space. The PCI Express Read Completions are then stored into the Endpoint IN FIFO. These transactions continue until the DMA Byte Count reaches 0. If the FIFO is full, the DMA Controller pauses until space is available. The USB Controller registers listed in Table 10-2 must be programmed for a single DMA Write transfer.

Writing to the channel's **DMASTAT** register *DMA Start* bit(s) (USB Controller, offset(s) 184h, 1A4h, 1C4h, 1E4h[0]) starts the DMA Data transfer. When the DMA Data transfer is complete, various interrupts are generated.

Table 10-2. IN Endpoints – Registers to Program for Single DMA Write Transfers

Offset(s)	Register	Description	Notes
180h, 1A0h, 1C0h, 1E0h	**DMACTL**	DMA Control	Select Single Transfer mode and other miscellaneous controls
190h, 1B0h, 1D0h, 1F0h	**DMACOUNT**	DMA Transfer Length	Set to the quantity of bytes and the direction to transfer
194h, 1B4h, 1D4h, 1F4h	**DMAADDR**	DMA Address	Set to the PCI Target address

10.3 Scatter/Gather Mode

In Scatter/Gather mode, creating a linked list of Descriptors can set up a series of DMA Data transfers. The list of Descriptors is stored in either on-chip, or PCI Express off-chip, memory, with the address of the first Descriptor programmed into the **DMADESC** register (USB Controller, offset(s) 198h, 1B8h, 1D8h, 1F8h). Each Descriptor consists of four DWords, located at offsets 0h, 4h, 8h, and Ch. (Refer to Table 10-3.) The register bits at offset 0h are described in the sections that follow.

Writing to the channel's **DMASTAT** register *DMA Start* bit(s) (USB Controller, offset(s) 184h, 1A4h, 1C4h, 1E4h[0]) starts the DMA Scatter/Gather Controller. The Controller then reads the four DWords at the PCI address, determined by the **DMADESC** register. A standard PCI Memory Read command is used for these transfers. These four DWords define the first DMA Data transfer, including the *Direction*, *DMA Byte Count*, *PCI Starting Address*, and *Next Descriptor Address* fields. The fourth DWord is divided into 16-bit ***Reserved*** and *User-Defined* fields, neither of which affects DMA Controller operation. After the DMA Data transfer completes, additional Descriptors are processed if the *End of Chain* bit is not Set.

Each Descriptor bit/field is described in the sections that follow.

Table 10-3. Scatter/Gather Mode Descriptors

Offset	31	30	29	28	27	26 25	24	23 4	3 0
0h	Valid	Direction	Done Interrupt Enable	End of Chain	DMA Scatter/ Gather FIFO Validate	DMA ISO Extra Transaction Opportunity	DMA OUT Continue	DMA Byte Count	
4h	PCI Starting Address								
8h	Next Descriptor Address								0000b
Offset	**31 16**					**15 0**			
Ch	***Reserved***					User-Defined			

10.3.1 *Valid* Bit

The *Valid* bit controls the processing of DMA Descriptors by the DMA Scatter/Gather Controller. When the firmware enters a Descriptor into the linked list, it Sets the *Valid* bit after the other fields in the Descriptor are written. As the DMA Scatter/Gather Controller processes the Descriptor list, it first checks the *Valid* bit, to determine whether a Descriptor is valid. If the *Valid* bit is Set, then Descriptor is considered valid and the corresponding DMA operation is started. If the *Valid* bit is Cleared, the Descriptor is periodically polled by the DMA Scatter/Gather Controller until the bit is Set. The polling rate is determined by the channel's **DMACTL** register *Descriptor Polling Rate* field(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[20:19]). The DMA Scatter/Gather Controller can be configured to pause when a Cleared *Valid* bit is detected. In this case, the firmware must restart the DMA Scatter/Gather Controller after setting up and validating additional Descriptors.

After a DMA Data transfer completes, the DMA Scatter/Gather Controller can be configured by the register's *DMA Clear Count Enable* bit(s) (bit 21) to Clear the *Valid* bit, effectively returning Descriptor ownership to the firmware.

If a Descriptor is encountered with the following bit/field values:

- *DMA Byte Count* field Cleared,
- *Valid* bit Set, and
- *End of Chain* bit Cleared,

the DMA Scatter/Gather Controller processes the next Descriptor in the chain.

Some applications do not require the use of the *Valid* bit. In those cases, the register's *DMA Valid Bit Enable* bit(s) (bit 17) is Cleared, allowing Descriptors to be processed without regard to the *Valid* bit.

10.3.2 *Clear Count Enable* Bit

The channel's **DMACTL** register(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h) determines whether the first Descriptor DWord is updated after the DMA Data transfer is complete. If this update is enabled, the *Valid* bit and *DMA Byte Count* fields are Cleared after the DMA Data transfer completes.

10.3.3 *Direction* Bit

When the *Direction* bit is Cleared, the DMA Controller transfers data from the USB Host to the PCI Express interface (OUT packets). When the *Direction* bit is Set, the DMA Controller transfers data from the PCI Express interface to the USB Host (IN packets).

10.3.4 *Done Interrupt Enable* Bit

When a DMA Data transfer associated with a Descriptor completes (*DMA Byte Count* reaches 0), an interrupt is generated if the *Done Interrupt Enable* bit is Set.

10.3.5 *End of Chain* Bit

The *End of Chain* bit in the first DWord indicates that there are no more Descriptors in the chain. If there are more Descriptors in the chain, the third DWord contains the next Descriptor address.

10.3.6 *DMA Scatter/Gather FIFO Validate* Bit

If the *DMA Scatter/Gather FIFO Validate* bit is Set for an IN endpoint, a USB short packet is automatically validated when the DMA Scatter/Gather Controller transfer completes. Therefore, each entry in a Scatter/Gather Descriptor list can be individually programmed to validate short packets. The channel's **DMACTL** register *DMA FIFO Validate* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[2]) takes precedence over this bit. If the *DMA FIFO Validate* bit is Set, all USB short packets are automatically validated at the end of each DMA Data transfer.

10.3.7 *DMA ISO Extra Transaction Opportunity* Field

The *DMA ISO Extra Transaction Opportunity* field is loaded into the endpoint's **GPEP[3:0/Out/In]_HS_MAXPKT** register *Additional Transaction Opportunities* field(s) (USB Controller, Index 20h, 30h, 40h, 50h, 60h, 70h, 80h, 90h[12:11]) when the DMA Count Descriptor word is loaded from the Descriptor list. The number in this field represents the total number of ISO packets transmitted during the next microframe. Do ***not*** use this feature for large FIFOs (2,048 bytes), nor small two-packet ISO transfers of 1,025 to 2,048 bytes.

10.3.8 *DMA OUT Continue* Bit

If the *DMA OUT Continue* bit is Set and a short OUT packet is received, the endpoint's **EP_STAT** register *NAK Packets* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[4]) is Cleared when the FIFO is emptied, and the DMA Scatter/Gather Controller proceeds to read the next DMA Descriptor.

Additionally, when the *DMA OUT Continue* bit is Set, the endpoint's **EP_RSP** register *NAK OUT Packets Mode Clear* bit(s) (USB Controller, offset(s) 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h[2]) must also be Set.

10.3.9 *DMA Byte Count* Field

The *DMA Byte Count* field determines the quantity of bytes to transfer. The maximum DMA Data transfer size is 16 MB. This field is Cleared at the end of the DMA Data transfer if the channel's **DMACTL** register *DMA Clear Count Enable* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[21]) is Set.

10.4 DMA OUT Transfer Completion

The endpoint's **EP_STAT** register *Short OUT Packet Done Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[6]) is Set when the following occurs:

- Register's *Short OUT Packet Received Interrupt* bit (bit 5) is Set, and
- OUT FIFO becomes empty

The consequences for firmware are as follows:

- *Short OUT Packet Done Interrupt* bit is not Set if firmware Clears the *Short OUT Packet Received Interrupt* bit before the FIFO becomes empty
- *Short OUT Packet Done Interrupt* bit is not Set if the FIFO is empty and the USB 3382 receives a Zero-Length packet (with no Data Payload)

There are two methods firmware can use to reliably determine when a DMA OUT transfer is complete. Both methods require the endpoint's **EP_RSP** register *NAK OUT Packets Mode Set* and *NAK OUT Packets Mode Clear* bit(s) (USB Controller, offset(s) 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h[10 and 2], respectively) to be Set. Setting the *NAK OUT Packets Mode Set* and *NAK OUT Packets Mode Clear* bit(s) causes the USB 3382 to Set the register's *NAK Packets Set* and *NAK Packets Clear* bit(s) (bits [15 and 7], respectively), as well as the endpoint's **EP_STAT** register *NAK Packets* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[4]), when a short OUT packet is accepted, preventing new packet data from entering the FIFO.

The simplest method for detecting DMA OUT transfer completion is to generate an interrupt on the *Short OUT Packet Received Interrupt* bit. In the interrupt service routine, firmware polls the endpoint's **EP_AVAIL** register(s) (USB Controller, offset(s) 310h, 330h/3F0h, 350h/410h, 370h/430h, 390h/450h), and reads the corresponding quantity of bytes from the endpoint's **EP_DATA** register(s) (USB Controller, offset(s) 314h, 334h/3F4h, 354h/414h, 374h/434h, 394h/454h). When the FIFO is empty, the endpoint's *FIFO Empty* status flag (**EP_STAT** register *FIFO Empty* bit(s) (bit 10)) is Set. The *Short OUT Packet Received Interrupt* bit is Set when the new packet enters the FIFO. Polling until the FIFO is empty ensures that newly arrived data is completely written to the PCI Express interface. In most PCI systems, the polling loop is short, because the DMA Controller bursts FIFO data to the PCI Express interface while the firmware is polling. Polling that takes more than a few loops to complete indicates that the DMA Data transfer is not working (*for example*, the target is not accepting DMA cycles).

To avoid polling, firmware can use a second method – generate an interrupt on the *Short OUT Packet Received Interrupt* or *Short OUT Packet Done Interrupt* bit. In the interrupt service routine, check whether the endpoint's **EP_AVAIL** register (or the *FIFO Empty* status flag) indicates that the FIFO is empty:

- If the FIFO is empty, the DMA Data transfer completed and standard OUT transfer completion handling can continue
- If the FIFO is not empty, firmware disables the *Short OUT Packet Received Interrupt* bit, and returns from the interrupt service routine

Firmware must not Clear the *Short OUT Packet Received Interrupt* bit at this time. Instead, firmware must disable the interrupt, by Clearing the endpoint's **EP_IRQENB** register *Short OUT Packet Received Interrupt Enable* bit(s) (USB Controller, offset(s) 308h, 328h/3E8h, 348h/408h, 368h/428h, 388h/448h[5]). When the DMA Data transfer completes and the FIFO is empty, the Short OUT Packet Done Interrupt occurs. The same interrupt service routine now finds the FIFO status to be empty; therefore, standard OUT transfer completion handling can continue.

10.5 DMA Scatter/Gather OUT Transfer Calculations

If a DMA Scatter/Gather OUT transfer is paused, the firmware might need to determine the quantity of bytes transferred up to that time.

If a DMA channel is paused before the current transfer completes, the **DMACOUNT**, **DMAADDR**, and **DMADESC** registers (USB Controller, offset(s) 190h, 1B0h, 1D0h, 1F0h; 194h, 1B4h, 1D4h, 1F4h; 198h, 1B8h, 1D8h, 1F8h, respectively) correspond to the current DMA Data transfer.

If the DMA channel is paused as the DMA Data transfer completes, the **DMADESC** register(s) is immediately updated to point to the next Descriptor; however, the **DMAADDR** and **DMACOUNT** registers are not updated until the next Descriptor is read. The window width varies, and is determined by other PCI Express interface activity that prevents the Descriptor from being read.

The situation is that when the DMA channel is paused, it is not known whether the **DMADESC** register(s) points to the current Descriptor or the next Descriptor.

If the **DMACOUNT** register(s) reflects a value of 0, the current DMA Data transfer completed and the **DMADESC** register(s) is pointing to the next DMA Descriptor. At this point, the new Descriptor is not read from memory.

If the **DMACOUNT** register(s) reflects a value equal to the *DMA Byte Count* field of the Descriptor pointed to by the **DMADESC** register(s), the previous DMA Data transfer is complete, and the **DMACOUNT**, **DMAADDR**, and **DMADESC** registers correspond to the new Descriptor. Otherwise, the three registers correspond to the current DMA Data transfer.

After these three registers are correlated, the firmware totals the quantity of bytes transferred in each Descriptor, up to and including the last Descriptor.

10.6 DMA Abort

If the CPU wants to stop the current DMA block transfer and the remainder of the DMA Descriptor transfers, it can abort the DMA channel by writing 1 to the channel's **DMASTAT** register *DMA Abort* bit(s) (USB Controller, offset(s) 184h, 1A4h, 1C4h, 1E4h[1]).

A DMA transfer is aborted when any of the following events occur:

- *DMA Abort* bit is Set
- Completion with a status other than Successful Completion is received
  - Completion with Unsupported Request status
  - Completion with Completer Abort status
- Completion timeout occurs
- Completion is malformed
- Completion has a Byte Count error
- DMA PCI Express Request hitting a Link downstream Port

When the *DMA Abort* bit is written, the DMA operation is terminated after finishing the current PCI Express TLP. The channel's **DMACTL** register *DMA Enable* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[1]) is Cleared. If a Scatter/Gather DMA Data transfer is in progress, no further Descriptor accesses are initiated. After the DMA Abort, flush the associated FIFO and re-program the DMA registers before attempting another DMA Data transfer.

10.7 DMA Pause

The USB 3382 supports two types of Pause, which are selected through the channel's **DMACTL** register *Pause Mode* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[23]):

- Graceful Pause Mode
- Immediate Pause Mode

The pause is triggered by writing 0 to the register's *DMA Enable* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[1]) when a DMA operation is in progress.

Both types of pause are described in the sections that follow.

10.7.1 Graceful Pause Mode

When the channel's **DMACTL** register *Pause Mode* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[23]) is Set and the register's *DMA Enable* bit(s) is (bit 1) Cleared, Descriptor fetching is disabled. DMA transfers associated with the current and prefetch Descriptors are allowed to finish. The channel's **DMASTAT** register *DMA Pause Done Interrupt* bit(s) (USB Controller, offset(s) 184h, 1A4h, 1C4h, 1E4h[26]) is Set, if enabled (**DMACTL** register *DMA Pause Done Interrupt Enable* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[26], is Set), when both Descriptors have finished. Information related to the last processed Descriptor is available in the registers.

If the *DMA Enable* bit is Set and the channel's **DMASTAT** register *DMA Start* bit(s) (USB Controller, offset(s) 184h, 1A4h, 1C4h, 1E4h[0]) is Set again by the CPU (8051 or PCI Express Host), the DMA simply resumes from where it is paused. In this case, Descriptor fetching is re-enabled and Data transfers are initiated. The next Descriptor is fetched from the location pointed to by the next Descriptor pointer. Software can change the next Descriptor pointer, before resuming (after being paused).

10.7.2 Immediate Pause Mode

When the channel's **DMACTL** register *Pause Mode and DMA Enable* bits (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[23 and 1], respectively) are both Cleared, Descriptor fetching is disabled, and no further PCI Express Requests are issued. The *DMA Pause Done Interrupt* bit(s) is Set, if enabled, as soon as any pending Requests are finished being issued and all Read Completions have been received. DMA operation is resumed from where it is paused, when the *DMA Enable* bit is Set again. The channel can also be aborted after an immediate pause.

The DMA Data transfer context is held while the channel is paused. When the *DMA Enable* bit is Set, the DMA Data transfer resumes from where it stopped. If the DMA Scatter/Gather Controller is active, it stops processing Descriptors during a DMA pause.

10.8 PCI Unaligned Write Transfers

DMA Write transfers are permitted to start at unaligned PCI address boundaries. *For example*, if the channel's **DMAADDR** register(s) (USB Controller, offset(s) 194h, 1B4h, 1D4h, 1F4h) is initialized to a value of 1 and there are three valid lines (12 bytes) in the OUT FIFO, the data is transferred. The channel's **DMAADDR** register value is 0000_000Dh after the DMA Write transfer completes.

There are no restrictions on unaligned DMA Write transfers.

10.9 PCI Unaligned Read Transfers

DMA Read transfers are allowed to start at unaligned PCI Address boundaries. *For example*, if the channel's **DMAADDR** register(s) (USB Controller, offset(s) 194h, 1B4h, 1D4h, 1F4h) is initialized to a value of 0000_0003h, and the channel's **DMACOUNT** register(s) (USB Controller, offset(s) 190h, 1B0h, 1D0h, 1F0h) is initialized to a value of 0000_0016h, the data is transferred. Bytes [3:0] are written to the first line in the FIFO, Bytes [7:4] to the second line, and so forth. The channel's **DMAADDR** register value is 0000_0013h after the DMA Read transfer completes.

There are no restrictions on unaligned DMA Read transfers.

Chapter 11 GPIO Controller Functional Description

11.1 GPIO Interface

The USB 3382 provides four General-Purpose I/O pins. GPIO pin functions are multiplexed with the LANE_GOOD[1:0]# pin and USB_LINK_GOOD# function. GPIO functions are selected when the **Debug Control** register *LANE_GOODx#/GPIOx Pin Function Select* bit (Port 0, offset 1DCh[22]) is Cleared, as described in Table 11-1.

As listed in Table 11-1, the USB 3382, by default, uses the GPIO0 signal for hardware control of external power regulators. However, to allow software control of the GPIO0/LANE_GOOD0# signal (*such as* to program I/O or read Lane status), software must additionally Set the **USB Power Management Control** register *GPIO0 Software Control* bit (USB Controller, offset 6C0h[11]).

When GPIO functions are selected, each GPIO pin is individually programmable through a set of **GPIO Control** registers, located in the USB Controller at offsets 50h through 60h. (Refer to Table 11-2.)

Table 11-1. GPIO Pin Multiplexing

GPIO Pin	Port 0, Offset 1DCh[22]=1	Port 0, Offset 1DCh[22]=0
GPIO0	LANE_GOOD0#	USB Power Management Function (default) (when **USB Power Management Control** register *GPIO0 Software Control* bit (USB Controller, offset 6C0h[11] is Cleared)
		GPIO0 (when **USB Power Management Control** register *GPIO0 Software Control* bit (USB Controller, offset 6C0h[11] is Set)
GPIO1	LANE_GOOD1#	GPIO1 (default)
GPIO2	No Function	GPIO2 (default)
GPIO3	USB_LINK_GOOD#	USB_LINK_ACTIVE (default) (when **GPIO Control** register *GPIO3 LED Select* bit (USB Controller, offset 50h[12] is Cleared)
		GPIO3 (when **GPIO Control** register *GPIO3 LED Select* bit (USB Controller, offset 50h[12] is Set)

Table 11-2. USB Controller GPIO Control Registers

Offset	Register
50h	**GPIO Control**
54h	**GPIO Status**
58h	**GPIO PWM Value**
5Ch	**GPIO PWM Ramp Control**
60h	**GPIO PWM Clock Frequency**

When a GPIO*x* pin is configured as an input, software can read the value presented on the pin. Additionally, the USB 3382 can generate an interrupt, based upon the pin's High or Low logic level. GPIO inputs can also be configured with hardware-based de-bounce circuitry, which helps prevent multiple interrupts for noisy or slow transitions on the signal pin.

When a GPIO*x* pin is configured as an output, software can program the pin to be High or Low. GPIO outputs also support programmable pulse-width-modulated (PWM) output, where software can program the quantity of Clock cycles that the signal drives High versus Low. This feature is useful for driving LEDs at various brightness levels, by varying the output waveform's Duty cycle. Additionally, the PWM Duty cycle can be linearly increased or decreased, at a programmable rate. This can be used to make LEDs progressively brighter or dimmer, at a constant rate.

PWM functions repeat in cycles of 256 steps. The duration (or period) of each step is determined by the **GPIO PWM Clock Frequency** register. The value in this register divides an input clock of 62.5 MHz by a programmed value of 0 to 128 (0 = 128). The PWM Clock Frequency value is used to divide a 62.5 MHz input clock. Allowable values are from 1 to 256 (00h to FFh, where 00h corresponds to 256). For the highest output frequency, program the register's *PWM Clock Divider* field (field [7:0]) to 01h. This yields a PWM frequency of 245.1 KHz. For the slowest output frequency (default), program the field to 00h, for a PWM frequency of 1.908 KHz. Values of 02h through FFh linearly scale the PWM frequency within this range. The PWM step time is common to all GPIO outputs.

PWM High time is controlled by way of the **GPIO PWM Value** register. Values programmed into this register determine the quantity of steps (out of 256) that the output is driven High. For the remaining steps in the PWM cycle, the output is driven Low.

The PWM Value for each GPIO can be incremented or decremented by one step every *n* PWM cycles, where *n* is the value programmed in the **GPIO PWM Ramp Control** register. PWM ramping starts when a value of *n*=1 to 255 is programmed, and ends when the PWM value reaches 255 or 0.

11.2 GPIO Control/Status Registers

GPIO registers are located only in the USB Controller. For details, refer to Section 16.5, "USB Controller Device-Specific Registers."

Chapter 12 I^{2}C/SMBus Slave Interface Operation

12.1 Introduction

This chapter discusses the I^{2}C Slave Interface and SMBus Slave Interface.

12.2 I^{2}C Slave Interface

12.2.1 I^{2}C Support Overview

Note: *This section applies to the I^{2}C Slave interface, which uses the I2C_ADDR[2:0], I2C_SCL, and I2C_SDA signals for USB 3382 register access by an I^{2}C Master.*

Inter-Integrated Circuit (I^{2}C) is a bus used to connect Integrated Circuits (ICs). Multiple ICs can be connected to an I^{2}C Bus, and I^{2}C devices that have I^{2}C mastering capability can initiate a Data transfer. I^{2}C is used for Data transfers between ICs at relatively low rates (100 Kbps), and is used in a variety of applications. For further details regarding I^{2}C Buses, refer to the *I2C Bus, v2.1*.

The USB 3382 is an I^{2}C Slave. Slave operations allow the USB 3382 Configuration registers to be read from or written to by an I^{2}C Master, external from the device. I^{2}C is a sideband mechanism that allows the device Configuration registers to be programmed, read from, or written to, independent of the PCI Express upstream Link.

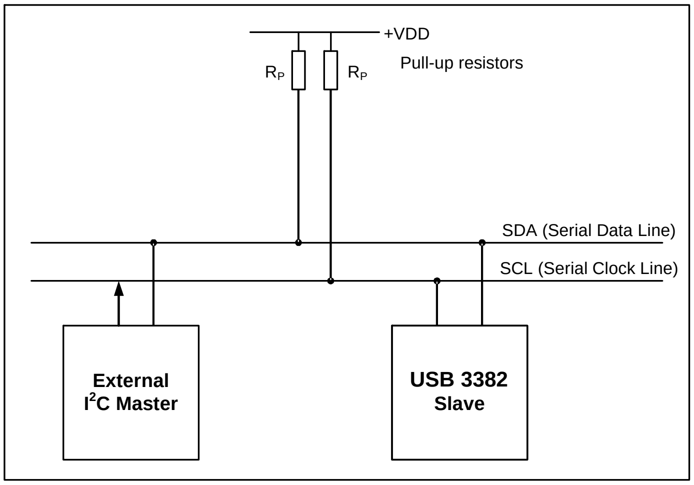
With I^{2}C, users have the option of accessing all USB 3382 registers through the I^{2}C Slave interface. I^{2}C provides an alternative to using a serial EEPROM. I^{2}C can also be used for debugging, such as if Port 0 fails to linkup.

Accordingly, it is recommended that both I^{2}C/SMBus access, and the serial EEPROM (or at least its footprint), be included in designs.

The I2C_SCL and I2C_SDA signals can be brought out to a 2x2 pin header on the board, to allow PLX software (*for example*, running on a laptop computer) to access the USB 3382 registers, using an Aardvark USB-I^{2}C adapter connected to this header. (Refer to the *USB 3382 RDK Hardware Reference Manual* for the header pin design.)

Figure 12-1 provides a block diagram that illustrates how standard devices connect to the I^{2}C Bus.

Figure 12-1. Standard Devices to I2C Bus Connection Block Diagram



12.2.2 I^{2}C Addressing – Slave Mode Access

To access the USB 3382 Configuration registers through the I^{2}C Slave interface, the USB 3382 I^{2}C Slave address must be configured.

The USB 3382 supports a 7-bit I^{2}C Slave address. The 7-bit I^{2}C Address bits can be configured by the serial EEPROM (recommended, if the default address must be changed), or by a Memory Write, in the **I^{2}C Configuration** register (Port 0, offset 294h, default value 5Fh), with the lower three bits of the address derived from the I2C_ADDR[2:0] inputs. Bits [6:0] correspond to Address Byte bits [7:1], with bit 0 of the byte indicating a Write (0) or Read (1).

The I2C_ADDR[2:0] inputs can be pulled or tied High or Low, to select a different Slave address. Up to eight USB 3382 devices can share the same I^{2}C Bus segment without conflict, provided that each USB 3382 has its I2C_ADDR[2:0] inputs strapped to a unique state. More than eight USB 3382 devices can share the I^{2}C Bus, however, if the upper Address bits are programmed in the serial EEPROM. The default state for I2C_ADDR[2:0] inputs that are not externally connected High or Low is 111, due to the internal pull-up resistors.

Note: *If the I2C_ADDR[2:0] inputs must be a value of 000, Address bits [2:0] must be externally driven Low, due to internal pull-up resistors.*

12.2.3 I^{2}C Slave Interface Register

The **I^{2}C Slave Interface** register, **I^{2}C Configuration** (Port 0, offset 294h), is described in Section 15.14.4, "Device-Specific Registers – I^{2}C and SMBus Slave Interfaces (Offsets 290h – 2C4h)." The default I^{2}C Slave address can be changed in the **I^{2}C Configuration** register to a different value, using the serial EEPROM or a Memory Write. **The I^{2}C Slave address must not be changed by an I^{2}C Write command.** (Refer to Section 12.2.2.)

Other I^{2}C Slave interface registers exist; however, they are for ***Factory Test Only***.

12.2.4 I^{2}C Command Format

An I^{2}C transfer starts as a packet with Address Phase bytes, followed by four Command Phase bytes, and one or more Data Phase bytes. The I^{2}C packet Address Phase Byte format is illustrated in Figure 12-2a. The Command Phase portion must include 4 bytes of data that contain the following:

- I^{2}C Transfer type (Read/Write)
- PCI Express Configuration Register address
- USB 3382 Port Number being accessed
- Byte Enable(s) of the register data being accessed

When the I^{2}C Master is writing to the USB 3382, the I^{2}C Master must transmit the Data bytes to be written to that register within the same packet that contains the Command bytes. Table 12-2 describes each I^{2}C Command byte for Write access. Figure 12-2b illustrates the Command phase portion of an I^{2}C Write packet.

When the I^{2}C Master is reading from the USB 3382, the I^{2}C Master must separately transmit a Command Phase packet and Data Phase packet. Table 12-6 describes each I^{2}C Command byte for Read access. Figure 12-4b illustrates the Command phase portion of an I^{2}C Read packet.

Each I^{2}C packet must contain 4 bytes of data. Pad unused packet Data bytes with zeros (0) to meet this requirement.

12.2.5 I^{2}C Register Write Access

The USB 3382 Configuration registers can be read from and written to, based upon I^{2}C register Read and Write operations, respectively. An I^{2}C Write packet consists of Address Phase bytes and Command Phase bytes, followed by one to four additional I^{2}C Data bytes. Table 12-1 defines mapping of the I^{2}C Data bytes to the Configuration register Data bytes. Figure 12-2c illustrates the I^{2}C Data byte format.

The I^{2}C packet starts with the *S* (START condition) bit. Data bytes are separated by the *A* (Acknowledge Control Packet (ACK)) or *N* (Negative Acknowledge (NAK)) bit. The packet ends with the *P* (STOP condition) bit.

If the Master generates an invalid command, the targeted USB 3382 register is not modified.

The USB 3382 considers the 1st Data byte of the 4-byte Data phase, following the four Command bytes in the Command phase, as register Byte 3 (bits [31:24]). The next three Data bytes access register Bytes 2 through 0, respectively. Four Data bytes are required, regardless of the Byte Enable Settings in the Command phase. The Master can then generate either a STOP condition (to finish the transfer) or a repeated START condition (to start a new transfer). If the I^{2}C Master sends more than the four Data bytes (violating USB 3382 protocol), the USB 3382 returns a NAK for the extra Data byte(s). (For further details regarding I^{2}C protocol, refer to the *I2C Bus, v2.1*.)

Table 12-2 describes each I^{2}C Command byte for Write access. In the packet described in Figure 12-2, Command Bytes 0 through 3 for Writes follow the format specified in Table 12-2.

Table 12-1. I^{2}C Register Write Access

I^{2}C Data Byte Order	PCI Express Configuration Register Bytes
0	Written to register Byte 3
1	Written to register Byte 2
2	Written to register Byte 1
3	Written to register Byte 0

Table 12-2. I^2C Command Format for Write Access

<table>
<tr><th>Byte</th><th>Bit(s)</th><th>Description</th></tr>
<tr><td rowspan="2">1st (0)</td><td>7:3</td><td>Reserved
Should be Cleared.</td></tr>
<tr><td>2:0</td><td>Command
011b = Write register
Do not use other encodings for Writes.</td></tr>
<tr><td rowspan="2">2nd (1)</td><td>7:2</td><td>Reserved
Should be Cleared.</td></tr>
<tr><td>1:0</td><td>Port Selector, Bits [2:1]
2nd Command byte, bits [1:0], and 3rd Command byte, bit 7, combine to form a 3-bit Port Selector.</td></tr>
<tr><td rowspan="4">3rd (2)</td><td>7</td><td>Port Selector, Bit 0
2nd Command byte, bits [1:0], and 3rd Command byte, bit 7, combine to form a 3-bit Port Selector. Port Selector[2:0] select the register block to access.
000b = Port 0 (Enhanced Adapter mode) / USB Controller PCI Configuration registers (Type 0) (Legacy Adapter mode)
001b = Port 1 (Enhanced Adapter mode)
010b = Port 2 (Enhanced Adapter mode)
011b = USB Controller PCI Configuration registers (Type 0) (Enhanced Adapter mode)
100b = USB Controller Configuration registers
All other encodings are reserved.</td></tr>
<tr><td>6</td><td>Reserved
Should be Cleared.</td></tr>
<tr><td>5:2</td><td>Byte Enables
Bit Description
2 Byte Enable for Byte 0 (USB 3382 register bits [7:0])
3 Byte Enable for Byte 1 (USB 3382 register bits [15:8])
4 Byte Enable for Byte 2 (USB 3382 register bits [23:16])
5 Byte Enable for Byte 3 (USB 3382 register bits [31:24])
0 = Corresponding USB 3382 register byte will not be modified
1 = Corresponding USB 3382 register byte will be modified
All 16 combinations are valid values.</td></tr>
<tr><td>1:0</td><td>USB 3382 Register Address [11:10]</td></tr>
<tr><td>4th (3)</td><td>7:0</td><td>USB 3382 Register Address [9:2]
Note: All register addresses are DWord-aligned. Therefore, Address bits [1:0] are implicitly Cleared, and then internally incremented for successive I²C byte Writes.</td></tr>
</table>

Figure 12-2. I²C Write Packet

Figure 12-2a I²C Write Packet Address Phase Bytes

1st Cycle			
START	7 6 5 4 3 2 1	0	ACK/NAK
S	Slave Address[7:1]	Read/Write Bit 0 = Write	A

Figure 12-2b I²C Write Packet Command Phase Bytes

Command Cycle							
7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK
Command Byte 0	A	Command Byte 1	A	Command Byte 2	A	Command Byte 3	A

Figure 12-2c I²C Write Packet Data Phase Bytes

Write Cycle								
7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	STOP
Data Byte 0 (to selected register Byte 3)	A	Data Byte 1 (to selected register Byte 2)	A	Data Byte 2 (to selected register Byte 1)	A	Data Byte 3 (to selected register Byte 0)	A	P

12.2.5.1 I2C Register Write Example

The following tables illustrate a sample I2C packet for writing the **MSI Upper Address** register (Port 0, offset 50h), with data 1234_5678h.

Note: *The USB 3382 has a default I2C Slave address [6:0] value of 5Fh, with the I2C_ADDR[2:0] inputs having a value of 111. The byte sequence on the I2C Bus, as listed in the following tables and figures, occurs after the START and before the STOP bits, by which the I2C Master frames the transfer.*

Table 12-3. I2C Register Write Access Example – 1st Cycle

Phase	Value	Description
Address	70h	**Bits [7:1] for USB 3382 I2C Slave Address (5Fh)** Last bit (bit 0) for Write = 0.

Table 12-4. I2C Register Write Access Example – Command Cycle

Byte	Value	Description	
0	03h	[7:3]	***Reserved*** Should be Cleared.
		[2:0]	**Command** 011b = Write register
1	00h	[7:2]	***Reserved*** Should be Cleared.
		[1:0]	**Port Selector, Bits [2:1]**
2	BCh	7	**Port Selector, Bit 0**
		6	***Reserved*** Should be Cleared.
		[5:2]	**Byte Enables** All active.
		[1:0]	**USB 3382 Register Address [11:10]**
3	14h	[7:0]	**USB 3382 Register Address [9:2]**

Table 12-5. I2C Register Write Access Example – Write Cycle

Byte	Value	Description
0	12h	**Data to Write for Byte 3**
1	34h	**Data to Write for Byte 2**
2	56h	**Data to Write for Byte 1**
3	78h	**Data to Write for Byte 0**

Figure 12-3. I²C Write Command Packet Example

Figure 12-3a I²C Write Packet Address Phase Bytes

1st Cycle			
START	7 6 5 4 3 2 1	0	ACK/NAK
S	Slave Address 1011_111b	Read/Write Bit 0 0 = Write	A

Figure 12-3b I²C Write Packet Command Phase Bytes

Command Cycle							
7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK
Command Byte 0 0000_0011b	A	Command Byte 1 0000_0000b	A	Command Byte 2 1011_1100b	A	Command Byte 3 0001_0100b	A

Figure 12-3c I²C Write Packet Data Phase Bytes

Write Cycle								
7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	STOP
Data Byte 0 0001_0010b	A	Data Byte 1 0011_0100b	A	Data Byte 2 0101_0110b	A	Data Byte 3 0111_1000b	A	P

12.2.6 I^2C Register Read Access

When the I^2C Master attempts to read a USB 3382 register, two packets are transmitted. The 1st packet consists of Address and Command Phase bytes to the Slave. The 2nd packet consists of Address and Data Phase bytes.

According to the *I2C Bus, v2.1*, a Read cycle is triggered when the Read/Write bit (bit 0) of the 1st cycle is Set. The Command phase reads the requested register content into the internal buffer. When the I^2C Read access occurs, the internal buffer value is transferred on to the I^2C Bus, starting from Byte 3 (bits [31:24]), followed by the subsequent bytes, with Byte 0 (bits [7:0]) being transferred last. If the I^2C Master requests more than four bytes, the USB 3382 re-transmits the same byte sequence, starting from Byte 3 of the internal buffer.

The 1st and 2nd I^2C Read packets (illustrated in Figure 12-4 and Figure 12-5, respectively) perform the following functions:

- **1st packet** – Selects the register to read
- **2nd packet** – Reads the register (sample 2nd packet provided is for a 7-bit USB 3382 I^2C Slave address)

Although two packets are shown for the I^2C Read, the I^2C Master can merge the two packets together into a single packet, by not generating the STOP at the end of the first packet (Master does not relinquish the bus) and generating REPEAT START.

Table 12-6 describes each I^2C Command byte for Read access. In the packet described in Figure 12-4, Command Bytes 0 through 3 for Reads follow the format specified in Table 12-6.

Table 12-6. I^{2}C Command Format for Read Access

Byte	Bit(s)	Description
1st (0)	7:3	***Reserved*** Should be Cleared.
	2:0	**Command** 100b = Read register Do not use other encodings for Reads.
2nd (1)	7:2	***Reserved*** Should be Cleared.
	1:0	**Port Selector, Bits [2:1]** 2nd Command byte, bits [1:0], and 3rd Command byte, bit 7, combine to form a 3-bit Port Selector.
3rd (2)	7	**Port Selector, Bit 0** 2nd Command byte, bits [1:0], and 3rd Command byte, bit 7, combine to form a 3-bit Port Selector. *Port Selector[2:0]* select the register block to access. 000b = Port 0 (Enhanced Adapter mode) / USB Controller PCI Configuration registers (Type 0) (Legacy Adapter mode) 001b = Port 1 (Enhanced Adapter mode) 010b = Port 2 (Enhanced Adapter mode) 011b = USB Controller PCI Configuration registers (Type 0) (Enhanced Adapter mode) 100b = USB Controller Configuration registers All other encodings are ***reserved***.
	6	***Reserved*** Should be Cleared.
	5:2	**Byte Enables** **Bit** **Description** 2 Byte Enable for Byte 0 (USB 3382 register bits [7:0]) 3 Byte Enable for Byte 1 (USB 3382 register bits [15:8]) 4 Byte Enable for Byte 2 (USB 3382 register bits [23:16]) 5 Byte Enable for Byte 3 (USB 3382 register bits [31:24]) 0 = Corresponding USB 3382 register byte will not be modified 1 = Corresponding USB 3382 register byte will be modified All 16 combinations are valid values.
	1:0	**USB 3382 Register Address [11:10]**
4th (3)	7:0	**USB 3382 Register Address [9:2]** ***Note:*** *All register addresses are DWord-aligned. Therefore, Address bits [1:0] are implicitly Cleared, and then internally incremented for successive I^{2}C byte Writes.*

Figure 12-4. I²C Read Command Packet (1st Packet)

Figure 12-4a I²C Read Command Packet Address Phase Bytes

1st Cycle			
START	7 6 5 4 3 2 1	0	ACK/NAK
S	Slave Address[7:1]	Read/Write Bit 0 = Write	A

Figure 12-4b I²C Read Command Packet Command Phase Bytes

Command Cycle								
7 6 5 4 3 2 1 0	ACK/ NAK	7 6 5 4 3 2 1 0	ACK/ NAK	7 6 5 4 3 2 1 0	ACK/ NAK	7 6 5 4 3 2 1 0	ACK/ NAK	STOP
Command Byte 0	A	Command Byte 1	A	Command Byte 2	A	Command Byte 3	A	P

Figure 12-5. I²C Read Data Packet (2nd Packet)

Figure 12-5a I²C Read Data Packet Address Phase Bytes

1st Cycle			
START	7 6 5 4 3 2 1	0	ACK/NAK
S	Slave Address[7:1]	Read/Write Bit, 1 = Read	A

Figure 12-5b I²C Read Data Packet Data Phase Bytes

Read Cycle								
7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	STOP
Register Byte 3	A	Register Byte 2	A	Register Byte 1	A	Register Byte 0	A	P

12.2.6.1 I2C Register Read Address Example – Phase and Command Packet

The following is a sample I2C packet for reading the **Serial EEPROM Buffer** register (Port 0, offset 264h), assuming the register value is ABCD_EF01h.

Note: *The USB 3382 has a default I2C Slave address [6:0] value of 5Fh, with the I2C_ADDR[2:0] inputs having a value of 111. The byte sequence on the I2C Bus, as listed in the following tables, occurs after the START and before the STOP bits, by which the I2C Master frames the transfer.*

Table 12-7. I2C Register Read Access Example – 1st Packet

Phase	Value	Description
Address	70h	**Bits [7:1] for USB 3382 I2C Slave Address (5Fh)** Last bit (bit 0) for Write = 0.

Table 12-8. I2C Register Read Access Example – Command Cycle

Byte	Value	Description
0	04h	[7:3] ***Reserved*** Should be Cleared. [2:0] **Command** 100b = Read register
1	00h	[7:2] ***Reserved*** Should be Cleared. [1:0] **Port Selector, Bits [2:1]**
2	BCh	7 **Port Selector, Bit 0** 6 ***Reserved*** Should be Cleared. [5:2] **Byte Enables** All active. [1:0] **USB 3382 Register Address [11:10]**
3	99h	[7:0] **USB 3382 Register Address [9:2]**

12.2.6.2 I²C Register Read Example – Data Packet

Note: *The USB 3382 has a default I²C Slave address [6:0] value of 5Fh, with the I2C_ADDR[2:0] inputs having a value of 111. The byte sequence on the I²C Bus, as listed in the following tables and figures, occurs after the START and before the STOP bits, by which the I²C Master frames the transfer.*

Table 12-9. I²C Register Read Access Example – 2nd Packet

Phase	Value	Description
Address	71h	**Bits [7:1] for USB 3382 I²C Slave Address (5Fh)** Last bit (bit 0) for Read = 1.
Read	ABh	**Byte 3 of Register Read**
	CDh	**Byte 2 of Register Read**
	EFh	**Byte 1 of Register Read**
	01h	**Byte 0 of Register Read**

Figure 12-6. 1st Packet – Command Phase

1st Cycle			
START	7 6 5 4 3 2 1	0	ACK/NAK
S	Slave Address 1011_111b	Read/Write Bit 0 = Write	A

Command Cycle								
7 6 5 4 3 2 1 0	ACK/ NAK	7 6 5 4 3 2 1 0	ACK/ NAK	7 6 5 4 3 2 1 0	ACK/ NAK	7 6 5 4 3 2 1 0	ACK/ NAK	STOP
Command Byte 0 0000_0100b	A	Command Byte 1 0000_0000b	A	Command Byte 2 1011_1100b	A	Command Byte 3 1001_1001b	A	P

Figure 12-7. 2nd Packet – Read Phase

1st Cycle			
START	7 6 5 4 3 2 1	0	ACK/NAK
S	Slave Address[7:1] 1011_111b	Read/Write Bit 1 = Read	A

Read Cycle								
7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	ACK/NAK	7 6 5 4 3 2 1 0	STOP	
Register Byte 3 1010_1011b	A	Register Byte 2 1100_1101b	A	Register Byte 1 1110_1111b	A	Register Byte 0 0000_0001b	P	

12.3 SMBus Slave Interface

12.3.1 SMBus Features

- Compliant to the *SMBus v2.0*
- Supports the SMBus Slave function only
- USB 3382 internal registers can be read and written, through the SMBus Slave interface
- Supports Address Resolution Protocol (ARP-capable)
- I2C_ADDR[2:0] inputs, serial EEPROM, software, or ARP Set the SMBus Device address
- Supports Block Read, Block Write, and Block Write - Block Read Process Call commands to access the registers
- Supports Packet Error Checking
- 10 to 100 KHz Bus operation frequency range

12.3.2 SMBus Operation

Based upon I^2C's principles of operation, SMBus is a two-wire bus used for communication between IC components and the remainder of the system. Electrically, I^2C and SMBus devices are compatible, and both protocol devices can co-exist on the same bus. Multiple devices, both Masters and Slaves, can be connected to an SMBus segment. PCI Express cards have two optional SMBus pins defined on the connector – SMCLK and SMDAT.

The USB 3382 implements an *SMBus v2.0*-compliant Slave device, and is used to read and write USB 3382 registers, through SMBus commands. The USB 3382 SMBus uses the same SDA data and SCL clock pins that are used for I^2C, and the I2C_ADDR[1:0] inputs, to define Device address assignment (I2C_ADDR2 is not used as an Address bit in SMBus mode). The SMBus Device Address is the same value as the I^2C Slave address, used by the I^2C protocol.

At any time, either the I^2C or the SMBus feature is enabled, dependent upon the **SMBus Configuration** register *SMBus Enable* bit (Port 0, offset 2ACh[0]) state, which is latched (at Fundamental Reset) to the inverse value of the STRAP_SMBUS_EN# input. Software can toggle this bit to switch between I^2C and SMBus functionality.

The USB 3382 SMBus Slave interface supports three command protocols for register access:

- Block Write
- Block Read
- Block Write - Block Read Process Call

The USB 3382 SMBus logic also supports the commands that are required to support ARP. ARP is a feature specific to *SMBus v2.0*, through which an SMBus ARP Master can dynamically assign a unique address to each of the SMBus Targets residing on the same bus. Although ARP is an optional feature of the *SMBus v2.0*, PCI and PCI Express cards are required to support ARP. The ARP feature is enabled when the **SMBus Configuration** register *ARP Disable* bit (Port 0, offset 2ACh[8]) is Cleared; this bit is initially latched (at Fundamental Reset) to the value of the I2C_ADDR2 input.

If ARP is disabled, by I2C_ADDR2 input being pulled High, the SMBus Slave Address bits [6:2] default to value 001_10b. Address bits [1:0] are initially latched (at Fundamental Reset) to the value of the I2C_ADDR[1:0] inputs, which allows a maximum of four SMBus-enabled USB 3382 to co-exist on the same SMBus segment. Software can change the SMBus Slave address, by programming the **SMBus Configuration** register *SMBus Device Address* field (Port 0, offset 2ACh[7:1]).

The USB 3382 also supports Packet Error Checking and Packet Error Code (PEC) generation, as explained in the *SMBus v2.0*. The *SMBus v2.0* optional feature, *Notify ARP Master* (which requires Master capability on the SMBus) is ***not*** supported.

12.3.3 SMBus Commands Supported

For register access, the SMBus logic supports three commands:

- Block Write (command BEh) is used to write the registers
- Block Write (command BAh), followed by Block Read (command BDh), can be used to read the registers
- Block Write - Block Read Process Call (commands BAh, CDh) can also be used to read registers

SMBus Commands that are not supported by the USB 3382 (Quick Command, Send Byte, Receive Byte, Write Byte, Write Word, Read Byte, Read Word, and Process Call), are NACKed.

12.3.3.1 SMBus Block Write

The Block Write command is used to write to the USB 3382 registers. General SMBus Block Writes are illustrated in Figure 12-8 and Figure 12-9. The sequence of Bytes include the following, in the sequence listed:

- 7-bit address,
- Command Code that indicates it is Block Write,
- *Byte Count* field with a value of 8h that indicates 4 bytes to set up the register to write (Port Number, register address, Command Byte Enable, and so forth), followed by
- 4 bytes of data to be written into the register

Figure 12-10 explains the elements used in Figure 12-8 and Figure 12-9, and Figure 12-11 indicates the Data Bytes written.

Figure 12-8. SMBus Block Write Command Format, to Write to a USB 3382 Register without PEC

S	Device Addr	Wr	A	Cmd code=BEh	A	Byte Count=8	A	Cmd Byte 1	A	Cmd Byte 2	A	Cmd Byte 3	A	

Cmd Byte 4	A	Data Byte 1	A	Data Byte 2	A	Data Byte 3	A	Data Byte 4	P	

Figure 12-9. SMBus Block Write Command Format, to Write to a USB 3382 Register with PEC

S	Device Addr	Wr	A	Cmd code=BEh	A	Byte Count=8	A	Cmd Byte 1	A	Cmd Byte 2	A	Cmd Byte 3	A

Cmd Byte 4	A	Data Byte 1	A	Data Byte 2	A	Data Byte 3	A	Data Byte 4	A	PEC	P

Figure 12-10. SMBus Packet Protocol Diagram Element Key

S -> START condition
P -> STOP condition

A -> Acknowledge (this bit position may be 0 for an ACK or 1 for a NACK)

[white box] -> Master to Slave
[grey box] -> Slave to Master

Figure 12-11. SMBus Block Write Bytes, as Written to Register

31:24	23:16	15:8	7:0
Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4

Note: *In each byte, the Most Significant Byte (MSB) is transmitted first.*

Table 12-10 provides a description of bytes for an SMBus Block Configuration Space register (CSR) Write.

Block Write transactions that are received with incorrect byte Settings are NACKed, starting from the wrong byte Setting, and including subsequent bytes in the packet. *For example*, if the Byte Count value is not 8, the USB 3382 NACKs the byte corresponding to the Byte Count value, as well as any Data bytes following within the same packet.

The byte after Data Byte 4, if present, is taken as the PEC byte, and if present, the PEC is checked. If a packet fails Packet Error Checking, the USB 3382 drops the packet (ignores the Write), and returns NACK for the PEC byte, to the SMBus Master. Packet Error Checking can be disabled, by Setting the **SMBus Configuration** register *PEC Check Disable* bit (Port 0, offset 2ACh[9]). The Byte Count value, by definition, does not include the PEC byte.

Table 12-10. Bytes for Block CSR Write on SMBus

Field (Byte) On Bus	Bit(s)	Value/Description
Command Code	7:0	BEh for **B**lock Writ**E**.
Byte Count	7:0	08h = 8 bytes to follow (4 Command and 4 Data bytes). The PEC byte is not counted.
Command Byte 1	7:3	***Reserved*** Should be Cleared.
	2:0	**Command** 011b = Write register 100b = Read register All other encodings are ***reserved***. ***Do not use.***
Command Byte 2	7:2	***Reserved*** Should be Cleared.
	1:0	**Port Selector, Bits [2:1]** 2nd Command byte, bits [1:0], and 3rd Command byte, bit 7, combine to form a 3-bit Port Selector.
Command Byte 3	7	**Port Selector, Bit 0** 2nd Command byte, bits [1:0], and 3rd Command byte, bit 7, combine to form a 3-bit Port Selector. *Port Selector[2:0]* select the register block to access. 000b = Port 0 (Enhanced Adapter mode) / USB Controller PCI Configuration registers (Type 0) (Legacy Adapter mode) 001b = Port 1 (Enhanced Adapter mode) 010b = Port 2 (Enhanced Adapter mode) 011b = USB Controller PCI Configuration registers (Type 0) (Enhanced Adapter mode) 100b = USB Controller Configuration registers All other encodings are ***reserved***.
	6	***Reserved*** Should be Cleared.
	5:2	**Byte Enables** **Bit** **Description** 2 Byte Enable for Byte 0 (USB 3382 register bits [7:0]) 3 Byte Enable for Byte 1 (USB 3382 register bits [15:8]) 4 Byte Enable for Byte 2 (USB 3382 register bits [23:16]) 5 Byte Enable for Byte 3 (USB 3382 register bits [31:24]) 0 = Corresponding USB 3382 register byte will not be modified 1 = Corresponding USB 3382 register byte will be modified All 16 combinations are valid values.
	1:0	**USB 3382 Register Address [11:10]**
Command Byte 4	7:0	**USB 3382 Register Address [9:2]** ***Note:*** *All register addresses are DWord-aligned. Therefore, Address bits [1:0] are implicitly Cleared, and then internally incremented for successive I²C byte Writes.*

Sample Register Write Byte Sequence Using SMBus Block Write

An SMBus Block Write packet to write to the **MSI Upper Address** [63:32] register (Port 2, offset 50h), is listed in Table 12-11. The register value is 1234_5678h, with all Bytes enabled, and without PEC. The default SMBus Device Address is 0111_000b.

Table 12-11. SMBus Sample Block Write Byte Sequence

Byte Number	Byte Type	Value	Description
1	Address	70h	Bits [7:1] for the USB 3382 default address is 38h, with bit 0 Cleared to indicate a Write.
2	Command Code	BEh	Command Code for register Write, using a Block Write.
3	Byte Count	08h	Byte Count. Four Command Bytes and Four Data Bytes.
4	Command Byte 1	03h	For Write command.
5	Command Byte 2	01h	Bits [7:2] are ***reserved***. Bits [1:0] – Port Selector [2:1].
6	Command Byte 3	3Ch	Bit 7 is Port Selector, bit 0. *Port Selector [2:0]*=010b for Port 2. Bit 6 is ***reserved***. Bits [5:2] are the four Byte Enables; all are active. Bits [1:0] are register Address bits [11:10].
7	Command Byte 4	14h	USB 3382 Register Address bits [9:2] (for offset 50h).
8	Data Byte 1	12h	Data MSB.
9	Data Byte 2	34h	Data Byte for register bits [23:16].
10	Data Byte 3	56h	Data Byte for register bits [15:8].
11	Data Byte 4	78h	Data LSB.

12.3.3.2 SMBus Block Read

A Block Read command is used to read USB 3382 registers. Similar to register Reads using I^2C, an SMBus Write sequence must first be performed to select the register to read, followed by an SMBus Read of the corresponding register. There are two ways a USB 3382 register can be read:

- Use a Block Write, followed by a Block Read. The Block Write sets up the parameters including Port Number, register address and Byte Enables, and the Block Read performs the actual Read operation.
- Use a Block Write - Block Read Process Call. This command is defined by the *SMBus v2.0*, and performs a Block Write and Block Read, using a single command. The Block Write portion of the Message sets up the register to be read, and then a repeated START followed by the Block Read portion of the Message returns the register data specified by the Block Write.

Note: *There is no STOP condition before the repeated START condition.*

CSR Read, Using SMBus Block Write, Followed by Block Read

A general SMBus Block Write and Block Read sequence is illustrated in Figure 12-12.

Table 12-12 describes the Byte definitions for a Block Write bus protocol, to prepare for a subsequent Block Read of the USB 3382 register.

The USB 3382 always NACKs any incorrect command sequences, starting with the wrong Byte. Upon receiving the Block Read command, the USB 3382 returns a PEC to the Master if, after the 4^{th} byte of register data, the Master still requests one more Byte. As a Slave, the USB 3382 recognizes the end of the Master's Read cycle, by observing the Master's NACK response for the last Data Byte transmitted by the USB 3382.

Incorrect command sequences are always NACKed, starting with the byte that is incorrect. (Refer to Table 12-13.) On the Block Read command, a PEC is returned to the Master, if after the 4^{th} byte of CSR data, the return Master still requests for one additional byte. As a Slave, the USB 3382 will know the end of the Master Read cycle, by observing the NACK for the last byte read from the Master.

Figure 12-12. SMBus Block Write to Set up Read, and Resulting Read that Returns CSR Value

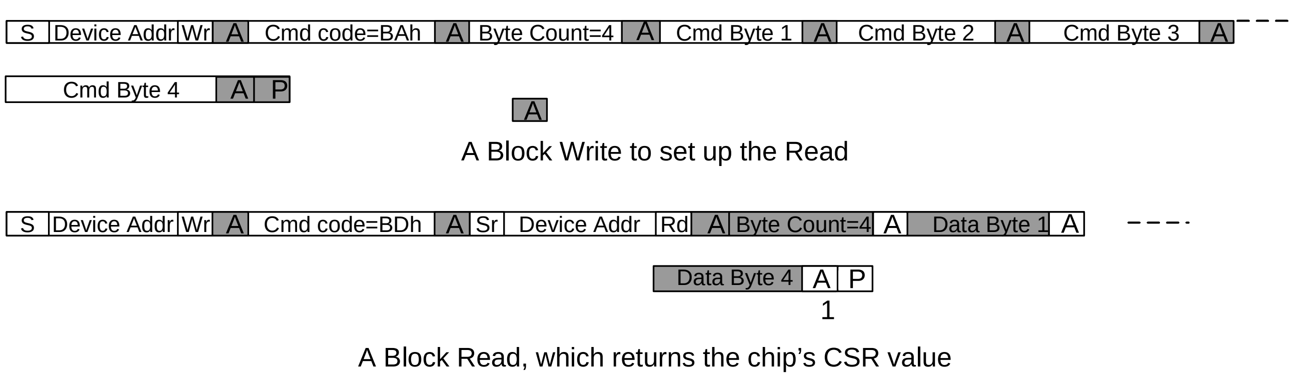


Table 12-12. SMBus Block Read Bytes

Field (Byte) On Bus	Bit(s)	Value/Description
Command Code	7:0	BAh, to set up the Read, using Block Writes.
Byte Count	7:0	04h = 4 Command bytes.
Command Byte 1	7:3	***Reserved*** Should be Cleared.
	2:0	**Command** 011b = Write register 100b = Read register All other encodings are ***reserved***. ***Do not use.***
Command Byte 2	7:2	***Reserved*** Should be Cleared.
	1:0	**Port Selector, Bits [2:1]** 2^{nd} Command byte, bits [1:0], and 3^{rd} Command byte, bit 7, combine to form a 3-bit Port Selector.
Command Byte 3	7	**Port Selector, Bit 0** 2^{nd} Command byte, bits [1:0], and 3^{rd} Command byte, bit 7, combine to form a 3-bit Port Selector. *Port Selector[2:0]* select the register block to access. 000b = Port 0 (Enhanced Adapter mode) / USB Controller PCI Configuration registers (Type 0) (Legacy Adapter mode) 001b = Port 1 (Enhanced Adapter mode) 010b = Port 2 (Enhanced Adapter mode) 011b = USB Controller PCI Configuration registers (Type 0) (Enhanced Adapter mode) 100b = USB Controller Configuration registers All other encodings are ***reserved***.
	6	***Reserved*** Should be Cleared.
	5:2	**Byte Enables** **Bit** **Description** 2 Byte Enable for Byte 0 (USB 3382 register bits [7:0]) 3 Byte Enable for Byte 1 (USB 3382 register bits [15:8]) 4 Byte Enable for Byte 2 (USB 3382 register bits [23:16]) 5 Byte Enable for Byte 3 (USB 3382 register bits [31:24]) 0 = Corresponding USB 3382 register byte will not be modified 1 = Corresponding USB 3382 register byte will be modified All 16 combinations are valid values.
	1:0	**USB 3382 Register Address [11:10]**
Command Byte 4	7:0	**USB 3382 Register Address [9:2]** ***Note:*** *All register addresses are DWord-aligned. Therefore, Address bits [1:0] are implicitly Cleared, and then internally incremented for successive* I^2C *byte Writes.*

Table 12-13. Command Format for SMBus Block Read

Field (Byte) On Bus	Bit(s)	Value/Description
Cmd Code	7:0	CDh, for Block Read (Process **C**all Rea**D**).

Sample CSR Read Byte Sequence, Using SMBus Block Write Followed by Block Read

An SMBus sequence to write and read the **MSI Upper Address** [63:32] register (Port 2, offset 50h), is listed in Table 12-14 and Table 12-15, respectively. The register value is ABCD_EF01h, and without PEC. The Block Write sets up the Port Numbers, Register address and Byte Enables, and the Block Read performs the real Read operation. The default SMBus Device Address is 0111_000b.

Table 12-14. SMBus Block Write Portion

Byte Number	Byte Type	Value	Description
1	Address	70h	Bits [7:1] value for the USB 3382 Slave address is 38h, with bit 0 Cleared to indicate a Write.
2	Block Write Command Code	BAh	Command Code for register Read setup, using a Block Write.
3	Byte Count	04h	Byte Count. Four Command Bytes.
4	Command Byte 1	04h	Write command.
5	Command Byte 2	01h	Bits [7:2] are ***reserved***. Bits [1:0] – Port Selector [2:1].
6	Command Byte 3	3Ch	Bit 7 is Port Selector, bit 0. *Port Selector [2:0]*=010b for Port 2. Bit 6 is ***reserved***. Bits [5:2] are the four Byte Enables; all are active. Bits [1:0] are register Address bits [11:10].
7	Command Byte 4	9Dh	USB 3382 Register Address bits [9:2] (for offset 50h).

Table 12-15. SMBus Block Read Portion

Byte Number	Byte Type	Value	Description
1	Address	70h	Bits [7:1] value for the USB 3382 Slave address is 38h, with bit 0 Cleared to indicate a Write.
2	Block Read Command Code	BDh	Command code for Block Read of USB 3382 registers.

Table 12-16. SMBus Read Command following Repeat START from Master

Byte Number	Byte Type	Value	Description
1	Address	71h	Bits [7:1] value for the USB 3382 Slave address is 38h, with bit 0 Set to indicate a Read.

Table 12-17. USB 3382 SMBus Return Bytes

Byte Number	Byte Type	Value	Description
1	Byte Count	04h	Four bytes in register.
2	Data Byte 1	ABh	Register data MSB.
3	Data Byte 2	CDh	Register data [23:16].
4	Data Byte 3	EFh	Register data [15:8].
5	Data Byte 4	01h	Register data LSB.

12.3.3.3 CSR Read, Using SMBus Block Write - Block Read Process Call

A general SMBus Block Write - Block Read Process Call sequence is illustrated in Figure 12-13. Alternatively, a general SMBus Block Write - Block Read Process Call with PEC sequence is illustrated in Figure 12-14.

Using this command, the register to be read can be set up and read back with one SMBus cycle (a transaction with a START and ending in STOP). There is no STOP condition before the repeated START condition. The command format for the Block Write part of this command has the same sequence as in Table 12-14, except that the Command Code changes to CDh, as illustrated below. Other Bytes remain the same as used in the sequence for SMBus Block Write followed by Block Read.

Table 12-13 lists the Command format for Block Read.

Figure 12-13. SMBus CSR Read Operation Using Block Write - Block Read Process Call

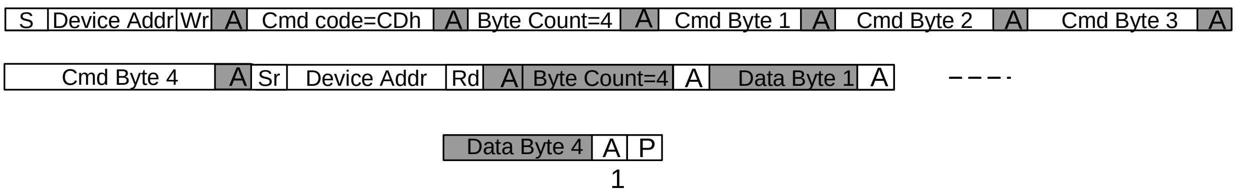
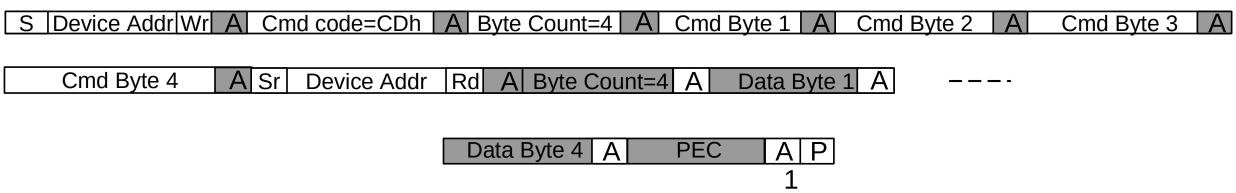


Figure 12-14. SMBus CSR Read Operation Using Block Write - Block Read Process Call with PEC



12.3.4 SMBus Address Resolution Protocol

Address Resolution Protocol (ARP) is a protocol by which SMBus devices that implement an assignable SMBus Device address feature are enumerated and dynamically assigned non-conflicting Slave addresses, rather than using a fixed Slave address. Although optional in the *SMBus v2.0*, it is mandatory per the *PCI r3.0* for add-in boards, to support ARP. This feature avoids conflicts with addresses used by other devices on a motherboard. ARP also allows multiple devices of the same type to co-exist on the same bus segment, without address conflicts.

To support this feature, a Slave device must implement a unique 128-bit ID, called *Unique Device Identifier (UDID)*. The fields of this ID are provided in Figure 12-15. All ARP commands use the default Device Address, 1100_001b. There are also two flags that the SMBus devices must implement to support the ARP process:

- **Address Resolved flag (AR)** – A flag bit or device internal state that indicates whether the ARP Master has resolved the device's SMBus Device address
- **Address Valid flag (AV)** – A flag bit or device internal state that indicates whether the device's SMBus Device address is valid

The process of assigning a SMBus Device address starts with the ARP Master issuing a Reset Device or Prepare to ARP command, using the default Device address. This Clears the *AR* flag in the Slave device (both flags are Cleared by a Reset Device command). The Master then issues a general Get UDID command. This causes all devices that support ARP to start driving their UDID onto the serial bus. A Target that loses the SMBus arbitration, backs off. Arbitration loss means that a device keeps the SMDAT line floating and it detects 0 driven by another device on the bus. Slave devices that lose arbitration issue NACK in response to further Bytes transmitted on the bus. After the ARP Master finishes the Get UDID sequence, it issues a Set Address command to the Slave device, using the Slave's UDID. All Slave devices on the bus monitor the UDID that is transmitted by the ARP Master, but only the particular device that has the matching UDID adopts the new SMBus Device address, and Sets its own *AV* and *AR* flags. After the Slave device Sets its *AR* flag, that device no longer responds to a general Get UDID command, which allows other devices to participate in the ARP process. All ARP commands require PEC checking and generation.

12.3.4.1 SMBus UDID

The 128-bit UDID is comprised of the following fields, as illustrated in Figure 12-15 (not to scale). Each UDID field and its default value implemented in the USB 3382 and meaning are explained in the tables that follow.

Figure 12-15. 128-Bit SMBus UDID

8 bits	8 bits	16 bits	16 bits	16 bits	16 bits	16 bits	32 bits
127:120	119:112	111:96	95:80	79:64	63:48	47:32	31:0
Device Capability	Version/ Revision	Vendor ID	Device ID	Interface	Subsystem Vendor ID	Subsystem Device ID	Vendor-Specific ID

Table 12-18. SMBus Vendor-Specific ID [31:0]

Field	Name	Default Value	Description
31:0	Vendor-Specific ID	Depends upon I2C_ADDR[2:0] input states. The four combinations provide the following ID values: 00b = 7000_0000h 01b = B000_0000h 10b = D000_0000h 11b = E000_0000h	The Vendor-Specific ID is used to provide a unique ID for functionally equivalent devices. This is for devices that would otherwise return identical UDIDs for the purpose of dynamic address assignment. The combination of two Address bits produces four unique Vendor-Specific ID values, for a maximum of four SMBus-enabled USB 3382s to co-exist on the same SMBus segment.

Table 12-19. SMBus Subsystem Device ID [47:32]

Field	Name	Default Value	Description
15:0	Subsystem Device ID	3382h	PLX part number for the USB 3382.

Table 12-20. SMBus Subsystem Vendor ID [63:48]

Field	Name	Default Value	Description
15:0	Subsystem Vendor ID	10B5h	PLX Vendor ID.

Table 12-21. SMBus Interface [79:64]

Field	Name	Default Value	Description
3:0	SMBus Version	0100b	*SMBus v2.0.*
15:4	***Reserved***	0-0h	Supported protocols.

Table 12-22. SMBus Device ID [95:80]

Field	Name	Default Value	Description
15:0	Device ID	3382h	USB 3382 default Device ID value.

Table 12-23. SMBus Vendor ID [111:96]

Field	Name	Default Value	Description
15:0	Vendor ID	10B5h	PLX Vendor ID.

Table 12-24. SMBus Version/Revision [119:112]

Field	Name	Default Value	Description
2:0	Silicon Revision ID	010b	USB 3382, Silicon Revisions AA and AB.
5:3	UDID Version	001b	UDID version defined for *SMBus v2.0*.
7:6	***Reserved***	00b	

Table 12-25. SMBus Device Capability [127:120]

Field	Name	Default Value	Description
0	PEC Supported	1	By default, PEC generation and checking are enabled.
5:1	***Reserved***	00_000b	
7:6	Address Type	10b	The USB 3382 SMBus Address Type is implemented as dynamic and volatile. 00b = Fixed address 01b = Dynamic and persistent 10b = Dynamic and volatile (default) 11b = Random number device

12.3.4.2 Supported SMBus ARP Commands

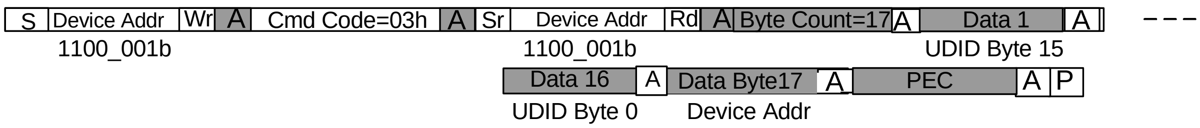
The USB 3382 supports all ARP Slave commands. The Notify ARP Master command, which requires Master functionality, is ***not*** supported. Table 12-26 explains the USB 3382 response to each received ARP command.

Table 12-26. SMBus ARP Commands Supported, Format, and Actions

ARP Command	SMBus Command Format	SMBus Device Address	Command Code	Action
Prepare to ARP (Only General)	Send Byte (Refer to Figure 12-16)	SMBus default Device Address 1100_001b	01h	Clear the *AR Flag* and prepare for the ARP process. *AV Flag* will have no change.
Reset Device (General)	Send Byte (Refer to Figure 12-16)	SMBus default Device Address 1100_001b	02h	Clear the *AR Flag* and *AV Flag*.
Reset Device (Directed)	Send Byte (Refer to Figure 12-16)	SMBus default Device Address 1100_001b	Target Device Address[7:1] + 0	If the *AV Flag* is Set, Set ACK and Clear the *AR Flag* and *AV Flag*; else, NACK/REJECT.
Get UDID (General)	Block Read (Refer to Figure 12-17)	SMBus default Device Address 1100_001b	03h	Respond only if the *AR Flag* is Cleared; else, NACK/REJECT. *AR Flag* and *AV Flag* are not changed. Address returned is all ones (1), if the *AV Flag* is Cleared.
Get UDID (Directed)	Block Read	SMBus default Device Address 1100_001b	Target Device Address[7:1] + 1	*AR Flag* and *AV Flag* are not changed. ACK if *AV Flag*=1; else, NACK/REJECT. Data Byte 17 returned will be the SMBus Device address.
Assign Address ARP	Block Write (Refer to Figure 12-18)	SMBus default Device Address 1100_001b	04h	Always ACK and Set the *AR Flag* and *AV Flag*, if the UDID matches.

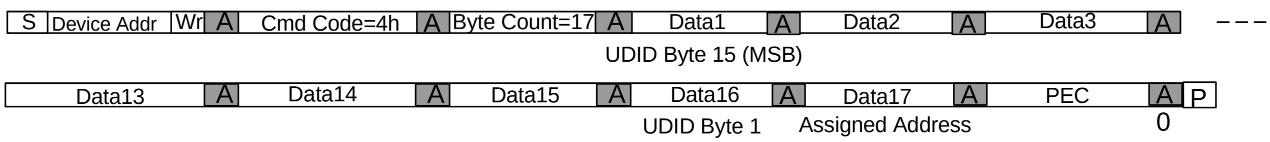
Figure 12-16. Prepare SMBus ARP Command and Reset Device Command Format

Figure 12-17. Get SMBus UDID Command Format (General Get UDID Command with PEC)



Note: *If the* ***SMBus Configuration*** *register AR Flag bit (Port 0, offset 2ACh[11]) is Cleared, the device returns the register's SMBus Device Address field (Port 0, offset 2ACh[7:1]) as 1111_111b; otherwise, it returns the device SMBus Device address. Bit 0 (LSB) in the Data Byte 17 field should be 1.*

Figure 12-18. Assign SMBus Address ARP Command Format



Note: *Bit 0 (LSB) of the Data 17 field is ignored in the Assign Address command field.*

12.3.5 SMBus PEC Handling

The USB 3382 supports the optional *SMBus v2.0* PEC generation and checking feature. This feature is required for the ARP process; however, it is optional for standard data transfer operation. The USB 3382 supports PEC Cyclic Redundancy Check (CRC) generation and checking during ARP, as well as during Read/Write transfers to USB 3382 registers. The CRC polynomial used for PEC calculation is:

$$C(x) = x^8 + x^2 + x + 1$$

An 8-bit parallel CRC is implemented. The PEC calculation does not include ACK, NACK, START, STOP, nor repeated START bits. An SMBus Master can determine whether a Slave device supports PEC, from the value of the UDID returned by the Slave device in response to a Get UDID command.

As a Slave device, the USB 3382 checks PEC, if the Master transmits the additional PEC byte and the USB 3382 PEC checking feature is enabled (default). PEC checking can be disabled, by Setting the **SMBus Configuration** register *PEC Check Disable* bit (Port 0, offset 2ACh[9]).

Additionally, when PEC is enabled, packets received with an incorrect PEC value are dropped. If PEC checking is disabled and a received PEC byte value is incorrect, the USB 3382 accepts the packet. During a register Read, if the Master requests the additional PEC byte, the USB 3382 generates and transmits the PEC byte after the register data.

12.3.6 Addressing USB 3382 SMBus Slave

By default, the USB 3382 supports ARP when the I2C_ADDR2 input is tied Low, and expects the ARP Master to define the USB 3382 SMBus Device address. If ARP is disabled by I2C_ADDR2 input being pulled High, the default address is 38h (Address bits [7:1] are 0111_000b, with Address bit [1:0] values loaded from the I2C_ADDR[1:0] inputs). The two Address bits allow a maximum of four USB 3382 SMBus Slaves to co-exist without address conflict on the SMBus, using SMBus address byte values of 70h, 72h, 74h, and 78h. The I2C_ADDR[2:0] inputs are loaded immediately after Fundamental Reset, and any subsequent change of input value does not affect functionality.

If the **SMBus Configuration** register *UDID Address Type* field is programmed as Fixed Address (Port 0, offset 2ACh[13:12], is Cleared) without disabling ARP, the USB 3382 still participates in ARP, but does not Set the Device address after ARP successfully completes.

The SMBus Slave Address can be changed at any time, by using software to write to the register's *SMBus Device Address* field (Port 0, offset 2ACh[7:1]). ARP can also be enabled or disabled at runtime, by writing to the register's *ARP Disable* bit (Port 0, offset 2ACh[8]). If ARP is disabled by software after initially being enabled, the default address (70h) is not used for subsequent transactions. In this case, software must program a Slave address into the *SMBus Device Address* field. When software writes the Device address, it must also Set the register's *AR Flag* and *AV Flag* bits (Port 0, offset 2ACh[11:10], respectively), to indicate that the address is valid and resolved.

Whenever software changes the register's *AV Flag*, *ARP Disable*, or *SMBus Device Address* values, software must also Set the register's *SMBus Parameter Re-Load* bit (Port 0, offset 2ACh[15]). Writes to this register bit take effect only when the register's *SMBus Command In-Progress* bit (Port 0, offset 2ACh[28]) is Cleared, which indicates that the USB 3382 SMBus interface is in the Idle state.

12.3.7 SMBus Timeout

Unlike I^{2}C, where the Slave or Master can indefinitely hold the I2C_SCL line Low, SMBus has a timeout condition. No device is allowed to hold the I2C_SCL line Low for more than 25 ms. When the USB 3382, as a slave-transmitter, detects that it has pulled the I2C_SCL line Low for more than 25 ms, the USB 3382 releases I2C_SCL, and the logic returns to its default state and waits for another START condition. This can also occur when the Master pulls the I2C_SCL line Low for more than 25 ms during single clock Low interval within a transfer in progress, or during the ACK phase if the Master pulls the I2C_SCL line Low to process a task. Generally, the USB 3382 pulls the I2C_SCL line Low if SMBus access to registers is delayed by internal arbitration for register access.

12.3.8 Switching between SMBus and I^{2}C Bus Protocols

The USB 3382's I^{2}C implementation allows switching between the SMBus and I^{2}C protocols, by toggling the **SMBus Configuration** register *SMBus Enable* bit (Port 0, offset 2ACh[0]).

When operating in SMBus mode, Clearing this bit, using the SMBus Block Write protocol, enables I^{2}C protocol for subsequent register accesses. This SMBus Block Write can be transmitted from an SMBus or I^{2}C Master, provided the Block Write Byte sequence conforms to the sequence explained in Section 12.3.3.1. In I^{2}C mode, writing 1 to the *SMBus Enable* bit turns On the SMBus protocol, immediately after the Write operation is complete.

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 13 Interrupt and Status Register Operation

13.1 Overview

This chapter describes, in general terms, how Interrupt and Status registers operate. For in-depth details of all Interrupt bits, refer to:

- Chapter 15, "PCI Configuration Registers"
- Chapter 16, "USB Configuration Registers"

For interrupt information specific to the GPIO*x* pins, refer to Chapter 11, "GPIO Controller Functional Description."

There are many sources of interrupts from the USB, PCI, DMA, and serial EEPROM sections of the USB 3382. Each of these interrupt sources can be routed to the internal 8051 or PCI Express Root Complex, by way of a Conventional PCI INT*x* interrupt or Message Signaled Interrupt (MSI). Each interrupt source consists of two *Enable* bits – one each for the 8051 and Conventional PCI INTA Message or MSI. When the USB 3382 is configured for Root Complex mode, the Conventional PCI INT*x* interrupt or MSI is written to the RCIN endpoint FIFO, and all interrupts are serviced by the 8051 or USB Host, by way of the STATIN Interrupt endpoint. The 8051 consists of two Interrupt inputs – one is asserted when an **IRQSTAT0** interrupt is active, and the other is asserted when an **IRQSTAT1** interrupt is active.

13.2 MSI Capability and Interrupt Request Status Registers

Interrupt sources in a Port/the SuperSpeed USB are grouped into two categories – **IRQSTAT1** and **IRQSTAT0** registers (USB Controller, offsets 2Ch and 28h, respectively). The **PCIIRQENB1** and **PCIIRQENB0** registers (USB Controller, offsets 14h and 10h, respectively) are used to enable individual Interrupt sources in **IRQSTAT1** and **IRQSTAT0**, respectively.

At configuration time, system software traverses the function Capability list. If a Capability ID of 05h is found, the function implements MSI. System software reads the **MSI Capability** Structure registers, to determine function capabilities.

The **MSI Control** register *Multiple Message Capable* field (All Ports and USB Controller, offset 48h[19:17]) default value is 001b, which indicates that the USB 3382 requests up to two MSI Vectors (Address and Data). When the register's *Multiple Message Enable* field (All Ports and USB Controller, offset 48h[22:20]) is Cleared (default), only one Vector is allocated, and therefore, the USB 3382 can generate only one Vector for all errors or events.

System software initializes the MSI Address registers (All Ports and USB Controller, offsets 4Ch and 50h) and **MSI Data** register (All Ports and USB Controller, offset 54h) with a system-specified Vector. After system software enables the MSI function (by Setting the **MSI Control** register *MSI Enable* bit (All Ports and USB Controller, offset 48h[16]), when an Interrupt event occurs, the Interrupt Generation module generates a DWord Memory Write to the address specified by the **MSI Address** (lower 32 bits of the *Message Address* field) and **MSI Upper Address** (upper 32 bits of the *Message Address* field) register contents (All Ports and USB Controller, offsets 4Ch and 50h, respectively). The single DWord Payload includes zero (0) for the upper two bytes, and the lower two bytes are taken from the **MSI Data** register.

The quantity of MSI Vectors that are generated is determined by the **MSI Control** register *Multiple Message Capable* and *Multiple Message Enable* fields (All Ports and USB Controller, offset 48h[19:17 and 22:20], respectively):

- If **one** MSI Vector is enabled (default mode), **IRQSTAT1** and **IRQSTAT0** interrupt events are combined into a single Vector
- If **two** MSI Vectors are enabled, the individual vectors indicate:
  - Vector[0] **IRQSTAT1**
  - Vector[1] **IRQSTAT0**

13.3 Endpoint Response Registers

Each Configurable endpoint has an **Endpoint Response** register (USB Controller, **EP_RSP** register, offset(s) 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h). This register determines how the USB 3382 responds to various situations during USB transactions. Writing 1 to the **EP_RSP** register's RW1C bits (bits [7:0]) Clears the corresponding bits. Writing 1 to the register's RW1S bits (bits [15:8]) Sets the corresponding bits. Reading Byte 0 or 1 of the register returns its current bit states.

13.4 Endpoint Status Registers

Each Configurable endpoint has an **Endpoint Status** register (USB Controller, **EP_STAT** register, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch). The register's bits are Set when a particular endpoint event occurs, and Cleared by writing 1 to the corresponding bit. **EP_IRQENB** register bits [6:5, 3:0] can cause an interrupt to be generated when the corresponding *Interrupt Enable* bits are Set. Reading the **EP_STAT** register returns its current bit states.

13.5 Message Signaled Interrupt Support on PCI Express Side

Other than Conventional PCI INT*x* interrupt messaging, the USB 3382 also supports Message Signaled Interrupts (MSIs) as an alternative when an interrupt is sent to PCI Express Host. Support for MSIs and INT*x* is mutually exclusive. If MSIs are enabled, INT*x* are automatically disabled, and vice versa. The function of MSIs are controlled through the **MSI Capability** structure registers. For in-depth details of the MSI registers, refer to Section 15.9, "MSI Capability Registers (Offsets 48h – 64h)."

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 14 Power Management

14.1 Overview

The USB 3382 supports *PCI Power Mgmt. r1.2* and *PCI Express Base r2.1*-compliant Power Management (PM), as well as the USB PM requirements detailed in the *USB r3.0*.

When operating in *USB r2.0* mode, the USB 3382 implements *USB r2.0* L1 (*USB r2.0* Link Power Management addendum) and L2 (Suspend state) Link PM states on the USB interface. When operating in *USB r3.0* SuperSpeed mode, the USB 3382 also implements the U1, U2, and U3 Link states. On the PCI Express interface, the USB 3382 implements ASPM L0s, ASPM L1, PCI Express L1, L2, and L2/L3 Ready Link PM states. Additionally, the USB 3382 supports the Conventional PCI-compatible PM D0, D1, D2, and D3 states.

This chapter covers the following PM topics:

- USB Power Configurations
- USB Interface Low-Power States
- PCI Express Power Management
- USB Suspend/Resume Sequences – Adapter Mode
- USB Suspend/Resume – Root Complex Mode
- USB Low-Power Suspend State

14.2 USB Power Configurations

The *USB r3.0* defines both bus- and self-powered devices:

- *Bus-powered* devices are peripherals that derive their power from the USB Host, by way of the USB connector
- *Self-powered* devices use an external power supply

The most significant consideration when deciding whether to build a bus- or self-powered device is power consumption. The *USB r3.0* dictates the following requirements for maximum current draw:

- Devices that are not Host-configured can draw only 100 mA from the USB connector Power pins
- Devices can draw no more than 500 mA from the USB connector Power pins
- In the Low-Power Suspend state, *USB r2.0* L2 and *USB r3.0* U3 Link states; however, devices can draw no more than 500 µA (low-power devices) or 2.5 mA (high-power devices) from the USB connector Power pins

Notice: If these power considerations can be met without using an external power supply, it is recommended that the USB 3382 be bus-powered; otherwise, implement a self-powered design.

14.2.1 Self-Powered Devices

Generally, a device with high-power requirements is self-powered. In a self-powered device, the USB 3382 and all other circuits are powered by the local power supply. This allows the local CPU to continue accessing the USB 3382, by way of the PCI Express interface, even when the USB 3382 is not connected to the USB Host. The USB connector's Power pin is connected only to the USB_VBUS input.

While connected to the USB interface, the USB 3382 automatically requests the Low-Power Suspend state, as described in Section 14.3.

14.3 USB Interface Low-Power States

14.3.1 *USB r2.0*-Compliant Power States

The USB 3382 supports the following *USB r2.0* low-power states:

- USB L1 Link PM State
- USB Suspend State or L2 Link PM State

14.3.1.1 USB L1 Link PM State

The USB 3382 supports the USB L1 Link PM state and can resume from it. The L1 Link PM state has exit latencies of 50 µs to 1 ms, compared to the L2 Link PM state, which has exit latencies of many milliseconds. The Host sends the extended token packet with the *bm* attributes. The attributes Set the Remote Wakeup feature. The USB 3382 returns an ACK, NYET, or STALL Handshake packet to the Host, depending upon the conditions explained:

- ACK, if L1 is supported and traffic conditions allow, –or–
- NYET, if there are pending packets, –or–
- STALL, if it does not understand the *bm* Request

The USB 3382 does not respond to L1 Link Power Management (LPM) Requests if the **USB2LPM** register *USB L1 LPM Support* bit (USB Controller, offset C0h[0]) is Cleared.

The USB 3382 supports Host resume or remote wakeup from the L1 Link PM state.

14.3.1.2 USB Suspend State or L2 Link PM State

When there is a 3-ms period of USB inactivity, the *USB r2.0* requires bus-powered devices to enter into the Low-Power Suspend state. During this state, bus-powered devices can draw no more 2.5 mA current. When in a Suspended state, the USB 3382 draws minimal current from the power supplies.

The USB 3382 supports Host-initiated resume and device remote wakeup signaling from the L2 Link PM state.

14.3.2 *USB r3.0*-Compliant Link States

The USB 3382 supports the following *USB r3.0* Link states:

- U1 Link State
- U2 Link State
- U3 Link State (USB Suspend State)

14.3.2.1 U1 Link State

U1 Link state entry can be initiated by either the USB 3382 or USB Host. The Host initiates U1 entry by sending an LGO_U1 Link command packet. The USB 3382 accepts or rejects this Request, depending upon traffic conditions. If ready to enter the U1 Link state, the USB 3382 accepts the Request by sending an LAU Link command, then waits for an LPMA Link command for entering the U1 Link state. Exiting the U1 Link state, to return to the U0 Link state, can be initiated by either the USB 3382 or USB Host.

The USB 3382 initiates U1 Link state entry only if the *U1_enable* flag is Set. This flag is Set by a Set Feature Setup packet with the *U1_ENABLE* field Set from the USB Host. This field is Cleared upon receipt of a Clear Feature Setup packet that has its *U1_ENABLE* field Set.

Even if the *U1_enable* flag is not Set, the USB 3382 accepts U1 Requests from the Host, if traffic conditions permit.

The U1 Link state has exit latencies of 1 to 2 µs.

14.3.2.2 U2 Link State

U2 Link state entry can be initiated by either the USB 3382 or USB Host. The Host initiates U2 entry by sending an LGO_U2 Link command packet. The USB 3382 accepts or rejects this Request, depending upon traffic conditions. If ready to enter the U2 Link state, the USB 3382 accepts the Request, by sending an LAU Link command, then waits for an LPMA Link command for entering the U2 Link state. Exiting the U2 Link state, to return to the U0 Link state, can be initiated by either the USB 3382 or USB Host.

The USB 3382 initiates U2 Link state entry only if the *U2_enable* flag is Set. This flag is Set by a Set Feature Setup packet with the *U2_ENABLE* field Set from the USB Host. This field is Cleared upon receipt of a Clear Feature Setup packet that has its *U2_ENABLE* field Set.

Even if the *U2_enable* flag is not Set, the USB 3382 accepts U2 Requests from the Host, if traffic conditions permit.

The U2 Link state has exit latencies of 80 to 100 µs (U2 has a higher exit latency than U1).

The USB 3382 can also directly enter the U2 Link state, after a U2 Inactivity timeout when in the U1 Link state. The U2 Inactivity timer is Set by a U2 Inactivity timeout LMP received from the USB Host.

14.3.2.3 U3 Link State (USB Suspend State)

The U3 Link state, also known as the *USB Suspend state*, is the lowest Link power state. U3 has a higher exit latency than the U1 and U2 Link states.

U3 Link state entry is always initiated by the USB Host, by sending an LGO_U3 Link command. The USB 3382 always accepts LGO_U3 Requests, by sending an LAU Link command, then settles into the U3 Link state after it receives an LPMA Link command.

Exit from the U3 Link state can be initiated either by the USB Host, or a device remote wakeup sent by the PCI Express Adapter (endpoint) device or PCI Express Host.

14.4 PCI Express Power Management

The USB 3382 provides Configuration registers and support hardware required by the *PCI Power Mgmt. r1.2* and PCI Express Link Power Management, as explained in the *PCI Express Base r2.1*.

The PCI-compatible device power states are programmed by writing into PCI power management control and status register, power state field. An interrupt, indicated by the **IRQSTAT1** register *Power State Change Interrupt Status* bit (USB Controller, offset 2Ch[27]), is generated when the power state is changed.

Table 14-1 lists supported and non-supported features and the register bits/fields used for configuration or activation. The USB 3382 also supports ASPM Link power states entry and exit, which is controlled autonomously by hardware, depending upon traffic conditions.

Table 14-1. Supported PCI Express PM Capabilities

Register		Description	Supported	
Offset	Bit(s)		Yes	No
40h		**PCI Power Management Capability (All Ports and USB Controller)**		
	7:0	**Capability ID** Program to 01h, to indicate that the Capability structure is the **PCI Power Management Capability** structure.	✔	
	15:8	**Next Capability Pointer** Default 48h points to the **MSI Capability** structure.	✔	
	18:16	**Version** Default 011b indicates compliance with the *PCI Power Mgmt. r1.2*.	✔	
	19	**PME Clock** Power Management Event (PME) clock. Does not apply to PCI Express. Returns a value of 0.		✔
	21	**Device-Specific Initialization** Default 0 indicates that Device-Specific Initialization is ***not*** required.	✔	
	24:22	**AUX Current** The **Data** register (All Ports and USB Controller, offset 44h[31:24]) is not implemented, by default. Until serial EEPROM and/or I^{2}C writes a value, the **Data** register field is all zeros (0s). If serial EEPROM and/or I^{2}C writes to the **Data** register, the **Data** register indicates that it is implemented, and those agents can then Clear the AUX Current value. If the **Data** register is implemented: **1.** This field returns a value of 000b. **2.** The **Data** register takes precedence over this field. If wakeup from the D3cold state is not supported, this field returns a value of 000b.	✔	
	25	**D1 Support** 1 = USB 3382 supports the D1 state	✔	
	26	**D2 Support** 1 = USB 3382 supports the D2 state	✔	
	31:27	**PME Support** Bits [31, 30, and 27] must be Set to indicate that the USB 3382 will forward PME Messages, as required by the *PCI Express Base r2.1*.	✔	

Table 14-1. Supported PCI Express PM Capabilities *(Cont.)*

Register Offset	Register Bit(s)	Description	Supported Yes	Supported No
44h		**PCI Power Management Status and Control (All Ports and USB Controller)**		
	1:0	**Power State** Used to determine the current Device PM state of the Port/SuperSpeed USB, and to program the Port/SuperSpeed USB into a new Device PM state. 00b = D0 01b = D1 10b = D2 11b = D3hot	✔	
	3	**No Soft Reset** 0 = D3hot to D0 state change causes a Fundamental Reset of the Port/SuperSpeed USB. This reset is propagated to downstream Ports and devices. 1 = Devices transitioning from the D3hot to D0 state, because of Power State commands, do not perform an internal reset.	✔	
	8	**PME Enable** 0 = Disables PME generation by the corresponding USB 3382 Port/SuperSpeed USB 1 = Enables PME generation by the corresponding USB 3382 Port/SuperSpeed USB	✔	
	12:9	**Data Select** Initially writable by serial EEPROM and/or I^2C only[a]. This Configuration Space register (CSR) access privilege changes to RW after a Serial EEPROM and/or I^2C Write occurs to this register. Selects the field [14:13] (*Data Scale*) and **PCI Power Management Data** register *Data* field (All Ports and USB Controller, offset 44h[31:24]). 0h = D0 power consumed 1h = D1 power consumed 2h = D2 power consumed 3h = D3 power consumed All other encodings are ***reserved***.	✔	
	14:13	**Data Scale** Writable by serial EEPROM and/or I^2C only[a]. Indicates the scaling factor to be used when interpreting the **PCI Power Management Data** register *Data* field (All Ports and USB Controller, offset 44h[31:24]) value. The value and meaning of the *Data Scale* field varies, depending upon which data value is selected by field [12:9] (*Data Select*). There are four internal *Data Scale* fields (one each, per *Data Select* values 0h, 1h, 2h, and 3h), per Port/SuperSpeed USB. For other *Data Select* values, the *Data* value returned is 00h.	✔	
	15	**PME Status** 0 = PME is not generated by the corresponding USB 3382 Port/SuperSpeed USB 1 = PME is being generated by the corresponding USB 3382 Port/SuperSpeed USB	✔	

*a. With no serial EEPROM nor previous I^2C programming, Reads return a value of 00h for the **PCI Power Management Status and Control** register Data Scale and **PCI Power Management Data** register Data fields (for all Data Selects).*

Table 14-1. Supported PCI Express PM Capabilities *(Cont.)*

Register Offset	Register Bit(s)	Description	Supported Yes	Supported No
44h		**PCI Power Management Control/Status Bridge Extensions (All Ports and USB Controller)**		
44h	22	**B2/B3 Support** ***Reserved*** Cleared, as required by the *PCI Power Mgmt. r1.2*.		✔
44h	23	**Bus Power/Clock Control Enable** ***Reserved*** Cleared, as required by the *PCI Power Mgmt. r1.2*.		✔
44h		**PCI Power Management Data (All Ports and USB Controller)**		
44h	31:24	**Data** Writable by serial EEPROM and/or I^2C only[a]. There are four supported *Data Select* values (0h, 1h, 2h, and 3h), per Port/ SuperSpeed USB. For other *Data Select* values, the *Data* value returned is 00h. Selected by the **PCI Power Management Status and Control** register *Data Select* field (All Ports and USB Controller, offset 44h[12:9]).	✔	
6Ch		**Device Capability (All Ports and USB Controller)**		
6Ch	8:6	**Endpoint L0s Acceptable Latency** 111b = Enables the capability	✔	
6Ch	11:9	**Endpoint L1 Acceptable Latency** 111b = Enables the capability	✔	
6Ch	25:18	**Captured Slot Power Limit Value** The upper limit on power supplied by the slot is determined by multiplying the value in this field by the value in field [27:26] (*Captured Slot Power Limit Scale*).	✔	
6Ch	27:26	**Captured Slot Power Limit Scale** The upper limit on power supplied by the slot is determined by multiplying the value in this field by the value in field [25:18] (*Captured Slot Power Limit Value*). 00b = 1.0 01b = 0.1 10b = 0.01 11b = 0.001	✔	

a. *With no serial EEPROM nor previous I^2C programming, Reads return a value of 00h for the* ***PCI Power Management Status and Control*** *register Data Scale and* ***PCI Power Management Data*** *register Data fields (for all Data Selects).*

Table 14-1. Supported PCI Express PM Capabilities *(Cont.)*

Register		Description	Supported	
Offset	Bit(s)		Yes	No
70h		**Device Control (All Ports and USB Controller)**		
	10	**AUX Power PM Enable**	✔	
		Device Status (All Ports and USB Controller)		
	20	**AUX Power Detected**	✔	
74h		**Link Capability (All Ports)**		
	11:10	**Active State Power Management (ASPM) Support** Active State Link PM support. Indicates the level of ASPM supported by the Port/ SuperSpeed USB. 01b = L0s Link PM state entry is supported 10b = L1 ASPM is supported 11b = L0s and L1 Link PM states are supported All other encodings are ***reserved***.	✔	
	14:12	**L1 Exit Latency** Indicates the L0s Link PM state exit latency for the given PCI Express Link. Value depends upon the **Advertised N_FTS** register *Advertised N_FTS* field (Port 0, offset B84h[7:0]) value and Link speed. Exit latency is calculated, as follows: • **2.5 GHz** – Multiply *Advertised N_FTS* x 4 (4 symbol times in 1 N_FTS) x 4 ns (1 symbol time at 2.5 GT/s) • **5.0 GHz** – Multiply *Advertised N_FTS* x 4 (4 symbol times in 1 N_FTS) x 2 ns (1 symbol time at 5.0 GT/s) 100b = Port/SuperSpeed USB's L0s Link PM state Exit Latency is 512 ns to less than 1 µs at 5.0 GT/s 101b = Port/SuperSpeed USB's L0s Link PM state Exit Latency is 1 µs to less than 2 µs at 2.5 GT/s All other encodings are ***reserved***.	✔	
	17:15	**L1 Exit Latency** Indicates the L1 Link PM state exit latency for the given PCI Express Link. Value depends upon the Link speed. 001b = Port/SuperSpeed USB's L1 Link PM state Exit Latency is 1 µs to less than 2 µs at 5.0 GT/s 010b = Port/SuperSpeed USB's L1 Link PM state Exit Latency is 2 µs to less than 4 µs at 2.5 GT/s All other encodings are ***reserved***.	✔	
	18	**Clock Power Management Capable**		✔

Table 14-1. Supported PCI Express PM Capabilities *(Cont.)*

Register Offset	Register Bit(s)	Description	Supported Yes	Supported No
78h		**Link Control (All Ports and USB Controller)**		
	1:0	**Active State Power Management (ASPM)** 00b = Disable[c] 01b = Enables only L0s Link PM state Entry 10b = Enables only L1 Link PM state Entry 11b = Enables both L0s and L1 Link PM state Entries	✔	
	8	**Clock Power Management Enable**		✔
138h		**Power Budget Extended Capability Header (All Ports and USB Controller)**		
	15:0	**PCI Express Extended Capability ID** Program to 0004h, as required by the *PCI Express Base r2.1*.	✔	
	19:16	**Capability Version** Program to 1h, as required by the *PCI Express Base r2.1*.	✔	
	31:20	**Next Capability Offset** Program to 148h, which addresses the **Virtual Channel Extended Capability** structure.	✔	
13Ch		**Data Select (All Ports and USB Controller)**		
	7:0	**Data Select** Indexes the Power Budget data reported, **Power Budget Data** registers, two per Port/ SuperSpeed USB, and selects the DWord of Power Budget data that appears in each **Power Budget Data** register. Index values start at 0, to select the first DWord of Power Budget data; subsequent DWords of Power Budget data are selected by increasing index values 0 to 1.	✔	

c. *The Port Receiver must be capable of entering the L0s Link PM state, regardless of whether the state is disabled.*

Table 14-1. Supported PCI Express PM Capabilities *(Cont.)*

Register Offset	Register Bit(s)	Description	Supported Yes	Supported No
		Power Budget Data (All Ports and USB Controller)		
140h	7:0	**Base Power** Two registers per Port/SuperSpeed USB. Specifies (in Watts) the base power value in the operating condition. This value must be multiplied by the field [9:8] (*Data Scale*) contents, to produce the actual power consumption value.	✔	
	9:8	**Data Scale** Specifies the scale to apply to the Base Power value. The device power consumption is determined by multiplying the field [7:0] (*Base Power*) contents with the value corresponding to the encoding returned by this field. 00b = 1.0x 01b = 0.1x 10b = 0.01x 11b = 0.001x	✔	
	12:10	**PM Sub-State** 000b = Power Management substate of the operating condition being described	✔	
	14:13	**PM State** Power Management state of the operating condition being described. 00b = D0 state 01b = D1 state 10b = D2 state 11b = D3 state	✔	
	17:15	**Type** Type of operating condition being described. 000b = PME Auxiliary 001b = Auxiliary 010b = Idle 011b = Sustained 111b = Maximum All other encodings are ***reserved***.	✔	
	20:18	**Power Rail** Power Rail of the operating condition being described. 000b = Power 12V 001b = Power 3.3V 010b = Power 1.8V 111b = Thermal All other encodings are ***reserved***.	✔	
		Note: *Two registers per Port/SuperSpeed USB can be programmed through the serial EEPROM, I^2C, and/or SMBus. Each non-zero register value describes the power usage for a different operating condition. Each configuration is selected by writing to the* ***Data Select*** *register Data Select field (All Ports and USB Controller, offset 13Ch[7:0]).*		

Table 14-1. Supported PCI Express PM Capabilities *(Cont.)*

Register Offset	Register Bit(s)	Description	Supported Yes	Supported No
144h		**Power Budget Capability (All Ports and USB Controller)**		
	0	**System Allocated** 1 = Power budget for the device is included within the system power budget	✔	
1E0h		**Power Management Hot Plug User Configuration (All Ports and USB Controller)**		
	0	**L0s Entry Idle Counter** Traffic idle time to meet, to enter the L0s Link PM state. 0 = Idle condition must last 1 µs 1 = Idle condition must last 4 µs	✔	
	7	**Disable PCI Express PM L1 Entry** 1 = Disables L1 Link PM state entry on Port 0, when Port 0 is placed into the D3hot state	✔	
	10	**L0s Entry Disable** 0 = Enables entry into the L0s Link PM state on a Port/the SuperSpeed USB when the L0s idle conditions are met 1 = Disables entry into the L0s Link PM state on a Port/the SuperSpeed USB when the L0s idle conditions are met	✔	
	12	**PME on Host Resume Enable** 1 = Enables forwarding of PME Messages when USB Host Resume signaling is detected on the USB interface while the USB 3382 is in the D1, D2, or D3hot state	✔	

14.4.1 Conventional PCI Device Power Management States

The PCI-compatible Device PM states, listed in Table 14-2, are programmed by writing into the **PCI Power Management Status and Control** register *Power State* field (All Ports and USB Controller, offset 44h[1:0]).

Table 14-2. Supported PCI Device Power Management States

Power State	Description
D0	Fully operational. This state requires the greatest amount of current.
D1	Light sleep. Only PCI Express Configuration transactions are accepted. The PCI Express interface requests the PCI Express L1 Link PM state when programmed to the D1 state.
D2	Heavy sleep. Same as the D1 state. Only PCI Express Configuration transactions are accepted. The PCI Express interface requests the PCI Express L1 Link PM state when programmed to the D2 state.
D3hot	Software accessible. By default, the USB 3382 does not need to be re-configured when programmed back from the D3hot-to-D0 state, without power and PEX_PERST# recycling. ***Note:*** *The USB 3382 is in this state before entering into a standby state or full power-off. Only PCI Express Configuration transactions are accepted. The PCI Express interface requests the PCI Express L1 Link PM state when programmed to the D3 state. After the USB 3382 is programmed to the D3hot state, sending a PME_Turn_Off Message to the Host can prepare the USB 3382 for turning Off the main power in Adapter mode.*
D3cold	Power Off. Main power to the USB 3382 is removed, with only the VAUX_CORE and VDD_IO supply rails optionally powered.

14.4.2 PCI Express Link Power Management States

The USB 3382 holds its upstream Link and downstream Links in the L0 Link PM state during standard operation (Conventional PCI-PM state is in the D0 Active state). ASPM defines a mechanism for components in the D0 state, to reduce Link power by placing their Links into a low-power state and instructing the other end of the Link to do likewise. This allows hardware-autonomous, dynamic Link power reduction beyond what is achievable by software-only-controlled PM. Table 14-3 defines the relationship between a component's Power state and upstream Link. Table 14-4 defines the relationship between Link PM states and power-saving actions.

Conventional PCI PM, and the L1 and L2/L3 Ready Link PM states are controlled by the PCI Express Root Complex programming the USB 3382 into the D3hot state, and subsequently broadcasting the PME_Turn_Off Message to the USB 3382.

The USB 3382 supports the following PCI Express Link PM states:

- ASPM L0s Link Power Management State
- ASPM L1 Link Power Management State
- L1 PCI Express Link Power Management State
- L2/L3 Ready PCI Express Link Power Management State
- L2 or L3 PCI Express Link Power Management State

Each is described in the sections that follow.

Table 14-3. Relationship between Component Power State and Upstream Link

Downstream Component State	Permissible Upstream Component State	Permissible Interconnect Link PM State
D0	D0	L0, L0s, L1 (optional) – ASPM.
D1	D0-to-D1	L1, L2/L3 Ready.
D2	D0-to-D2	L1, L2/L3 Ready.
D3hot	D0-to-D3hot	L1, L2/L3 Ready.
D3cold (Vaux)	D0-to-D3cold	L2 (Vaux present), L3 (Vaux not present).

Table 14-4. Relationship between Link PM States and Power-Saving Actions

Link Power Management State	Power-Saving Actions
Tx L0s	PCI Express Tx Lanes are in a High-Impedance state.
Rx L0s	PCI Express Rx Lanes are in a low-power state.
L1	PCI Express Tx and Rx Lanes are in a low-power state. FC timers are suspended. Can optionally turn off the USB 3382 internal clocks.
L2/L3 Ready	
L2 (D3cold with Vaux present)	Only WAKE# and beacon detect logic active with Vaux supply. Otherwise, the remainder of the USB 3382 is powered Off.
L3 (D3cold)	Component is fully powered Off. Vaux is not present.

14.4.2.1 ASPM L0s Link Power Management State

ASPM L0s Link PM state entry is controlled by the PCI Express Host programming the **Link Control** register *Active State Power Management (ASPM)* field (All Ports and USB Controller, offset 78h[1:0]) for ASPM control. When enabled and traffic idle conditions are met, the USB 3382 enters the L0s Link PM state, and the PHY Tx Lanes are driven to a low-power Electrical Idle state. Receivers go to the L0s Link PM state, even if the Transmitter is not enabled for L0s. Exit from Tx L0s is triggered by a pending DLLP or TLP to transmit.

In Root Complex mode, the downstream Port L0s Link PM state can be enabled by the USB Host, I^2C, serial EEPROM, and/or 8051 programming the *Active State Power Management (ASPM)* field for ASPM control.

14.4.2.2 ASPM L1 Link Power Management State

ASPM L1 Link PM state entry is controlled by the PCI Express Host programming the **Link Control** register *Active State Power Management (ASPM)* field (All Ports and USB Controller, offset 78h[1:0]) for ASPM control. When enabled and traffic idle conditions are met, the USB 3382 enters the L1 Link PM state after negotiation. Port 0 always starts L1 Link PM state negotiation. The PHY Tx and Rx Lanes are in a low-power Electrical Idle state. Exit from ASPM L1 is triggered by a pending DLLP or TLP to transmit, and can be triggered by either side of the Link.

In Root Complex mode, ASPM L1 on downstream Ports can be enabled by the USB Host, I^2C, serial EEPROM, and/or 8051 programming the *Active State Power Management (ASPM)* field for ASPM control. The downstream Ports enter the ASPM L1 state when a downstream PCI Express Adapter (endpoint) requests the ASPM L1 Link PM state, and successfully negotiates L1 Link PM state entry.

14.4.2.3 L1 PCI Express Link Power Management State

L1 Link PM state entry is controlled by the PCI Express Host programming the **PCI Power Management Status and Control** register *Power State* field (All Ports and USB Controller, offset 44h[1:0]) to a non-zero value. When enabled and traffic idle conditions are met, Port 0 enters the PCI Express PM L1 Link power state after L1 negotiation. Port 0 always starts L1 Link PM state negotiation in Adapter mode. The PHY Tx and Rx Lanes are in a low-power Electrical Idle state. Exit from PCI Express PM L1 is triggered by a pending DLLP or TLP to transmit, and it can be initiated by either side of the Link.

In Root Complex mode, PCI PM L1 state entry is initiated by the PCI Express endpoint device when the the USB Host and/or 8051 programs its *Device Power State* field to a non-zero value.

14.4.2.4 L2/L3 Ready PCI Express Link Power Management State

L2/L3 Ready Link PM state entry is controlled by a PCI Express Host sending a PME_Turn_Off Message to the USB 3382. When a PME_Turn_Off Message is received and traffic conditions are met, Port 0 sends a PME_TO_Ack Message and starts L2/L3 Ready Link PM state negotiation. The PHY Tx and Rx Lanes are in a low-power Electrical Idle state when in the L2/L3 Ready Link PM state.

In Adapter mode, when the PCI Express upstream Link enters the L2/L3 Ready Link PM state, the USB interface disconnects from the USB Host, except when the USB interface is in the Suspend state. When suspended, the USB interface is not disconnected and facilitates either USB Host wakeup or USB 3382 device remote wakeup. Additionally, a downstream Port enters the L2/L3 Ready Link PM state after a PCI Express Adapter (endpoint) behind it successfully negotiates to the L2/L3 Ready Link PM state.

In Root Complex mode, a PME_Turn_Off Message can be generated, using the PCIOUT Dedicated endpoint, to put the PCI Express endpoint devices and USB 3382 downstream Port Lanes into the L2/L3 Ready Link PM state.

14.4.2.5 L2 or L3 PCI Express Link Power Management State

The USB 3382 has the option of supporting a very low auxiliary power-operated Link/Device PM state when the PCI Express Link enters the L2/L3 Ready Link PM state in Adapter mode, or the USB Host is in the Suspend state in Root Complex mode. In Adapter mode, when the connected PCI Express system supports wakeup from the D3cold state and Auxiliary power is available, the USB 3382 settles into the D3cold state when main power is removed. The PCI Express Link settles into the L2 Link PM state.

If an Auxiliary power supply is not provided, and the system does not support the D3cold state, the PCI Express Link settles into the L3 Link PM state (powered Off). Refer to Section 14.7 for further details.

14.5 USB Suspend/Resume Sequences – Adapter Mode

The USB 3382 supports the following USB Suspend/Resume sequences in Adapter mode:

- Suspend Sequence with 8051 Held in Reset
- Host-Initiated Wakeup with 8051 Held in Reset
- Suspend Sequence with 8051 Operating
- Host-Initiated Wakeup with 8051 Operating

14.5.1 Suspend Sequence with 8051 Held in Reset

1. When a USB Host Suspend condition is detected, the USB 3382 Sets the **IRQSTAT1** register *Suspend Request Change Interrupt Status* bit (USB Controller, offset 2Ch[2]), and generates an Interrupt Message to the PCI Express Host if the corresponding interrupt is enabled in the **PCIIRQENB1** register *Suspend Request Change PCI Express Interrupt Enable* bit (USB Controller, offset 14h[2]).

2. The PCI Express Host places the USB 3382 into the low-power D1, D2, or D3hot state, by writing to the **PCI Power Management Status and Control** register (All Ports and USB Controller, offset 44h). If the **USBCTL** register *Immediately Suspend* bit (USB Controller, offset 8Ch[7]) is Set, after 500 µs the USB 3382 enters the Suspend state. If the *Immediately Suspend* bit is Cleared, the PCI Express Host must write to the **IRQSTAT1** register *Suspend Request Interrupt Status* bit (USB Controller, offset 2Ch[3]), then write to the **PCI Power Management Status and Control** register, to change the USB 3382 into the D1, D2, or D3hot state. After the PCI Express Host writes to the *Suspend Request Interrupt Status* bit, and after 500 µs has passed, if the PCI Express Link enters the PCI Express L1 Link PM state, the USB 3382 enters the Suspend state.

Note: *A Device-Remote Wakeup event is not recognized during the 500-µs suspend delay period.*

If a USB device is self-powered (*that is*, not drawing power from the USB Host by way of the USB VBUS line), the PCI Express Host can ignore the Suspend Request Change interrupt and does not need to place the USB 3382 into the D1, D2, or D3hot state.

14.5.2 Host-Initiated Wakeup with 8051 Held in Reset

The USB Host can wake up the USB 3382, by driving a non-idle state on the USB_DM/USB_DP signals when operating in *USB r2.0* speed, or by driving an LFPS that meets *USB r3.0* SuperSpeed U3 Link state exit requirements. The USB 3382 detects the Host's Wakeup request, then restarts its internal clock if it is already in a stopped state. The Host-Initiated Wakeup is recognized only when the USB_VBUS input is High, and the **USBCTL** register *USB Root Port Wakeup Enable* and *USB Detect Enable* bits (USB Controller, offset 8Ch[11 and 3]) are both Set. The USB 3382 asserts the Suspend Request Change interrupt to the PCI Express Host, with the **USBSTAT** register *Suspend Status* bit (USB Controller, offset 90h[9]) Cleared. The PCI Express Host then places the USB 3382 into the D0 state, where it is ready to process USB packets.

If the PCI Express Host must wakeup the USB 3382 in the absence of USB Host activity, the PCI Express Host can change the state to D0. The PCI Express Host can write 1 to the **USBSTAT** register *Generate Resume* bit (USB Controller, offset 90h[5]). This causes a device remote wakeup to the USB Host, if remote wakeup is enabled by the USB Host through the corresponding SET_FEATURE Request. When operating in *USB r3.0* SuperSpeed mode, and device remote wakeup is detected when in the U3 state, after the Link is brought back to the U0 state, the USB 3382 sends a Device remote wakeup Device notification TP toward the USB Host.

14.5.3 Suspend Sequence with 8051 Operating

1. When a USB Host Suspend condition is detected, the 8051 and PCI Express Host receive the Suspend Request Change interrupt.
2. The 8051 or PCI Express Host places the USB 3382 into the D1/D2/D3 state, by writing to the **PCI Power Management Status and Control** register (All Ports and USB Controller, offset 44h). If the **USBCTL** register *Immediately Suspend* bit (USB Controller, offset 8Ch[7]) is Set after 500 µs, the USB 3382 enters a Suspend state when PCI Express Link settles into the L1 Link PM state.

Note: *A Device-Remote Wakeup event is not recognized during the 500 µs suspend delay period.*

If a device is self-powered, the 8051 or PCI Express Host can ignore the Suspend Request Change interrupt, and never place the USB 3382 into the D1/D2/D3 state.

14.5.4 Host-Initiated Wakeup with 8051 Operating

The USB Host can wakeup the USB 3382, by driving a non-idle state on the USB interface when operating at *USB r2.0* speed or by sending an LFPS that meets *USB r3.0* SuperSpeed U3 Link state exit requirements. The USB 3382 detects the Host's Wakeup Request, then restarts its internal clocks. The Host-Initiated Wakeup is recognized only when the USB_VBUS input is High, and the **USBCTL** register *USB Root Port Wakeup Enable* and *USB Detect Enable* bits (USB Controller, offset 8Ch[11, 3]) are both Set. The 8051 and PCI Express Host receive the Resume interrupt, if the corresponding interrupt(s) is enabled. The 8051 or PCI Express Host then changes the USB 3382 power state to the D0 state, where it is ready to process USB packets.

14.6 USB Suspend/Resume – Root Complex Mode

The USB 3382 supports the following USB Suspend/Resume sequences in Root Complex mode:

- Suspend Sequence
- USB Host-Initiated Wakeup
- Device-Remote Wakeup
- Resume Interrupt

14.6.1 Suspend Sequence

If the 8051 is running, perform the following Suspend procedure when operating in Root Complex mode:

1. During Configuration register initialization, the **CPUIRQENB1** register *Suspend Request Change 8051 Interrupt Enable* bit (USB Controller, offset 1Ch[2]) is Set, to generate an 8051 interrupt.
2. When the USB is idle for 3 ms for the *USB r2.0* mode of operation, or the USB Link enters the U3 Link state in the *USB r3.0* SuperSpeed mode of operation, the USB 3382 Sets the **IRQSTAT1** register *Suspend Request Change Interrupt Status* bit (USB Controller, offset 2Ch[2]), which generates an interrupt to the 8051. This interrupt also occurs in *USB r2.0* mode when the USB 3382 is not connected to a Host, and the USB data lines are pulled to the idle state (HSDP and FSDP are High, and HSDM and FSDM are Low).

   This interrupt only occurs if the USB_VBUS input is High.
3. The 8051 accepts this interrupt by Clearing the *Suspend Request Change Interrupt Status* bit, and performs the required tasks. This can include placing other PCI Express devices into the Low-Power state and stopping the PCI Express clock to other devices.
4. The 8051 writes a 1 to the **IRQSTAT1** register *Suspend Request Interrupt Status* bit (USB Controller, offset 2Ch[3]), to initiate the transition to the Low-Power Suspend state. After 500 µs, USB 3382 enters the Low-Power Suspend state.

Note: *A Device-Remote Wakeup event is not recognized during the 500-µs suspend delay period.*

If the **USBCTL** register *Immediately Suspend* bit (USB Controller, offset 8Ch[7]) is Set, the USB 3382 automatically suspends when the USB Host is idle for 3 ms in *USB r2.0* mode, or the Link settles into the U3 Link state when operating in *USB r3.0* mode. After 500 µs, the USB Link enters a Suspend state that consumes very low power, if the PCI Express Link is already in the L1 Link PM state.

If a device is self-powered, it can ignore the USB Suspend Request and never write 1 to the *Suspend Request Interrupt Status* bit.

14.6.2 USB Host-Initiated Wakeup

The USB Host can wakeup the USB 3382, by driving a non-idle state on the USB interface when operating at *USB r2.0* speed or by driving an LFPS that meets *USB r3.0* SuperSpeed U3 Link state exit requirements. The USB 3382 detects the Host's Wakeup Request. The Host-Initiated Wakeup is recognized only when the USB_VBUS input is High, and the **USBCTL** register *USB Root Port Wakeup Enable* and *USB Detect Enable* bits (USB Controller, offset 8Ch[11, 3]) are both Set. The 8051 receives a Resume interrupt and powers up the other PCI Express devices and associated clocks. The USB 3382 GPIO pins can be used for controlling power and clocks to other devices.

14.6.3 Device-Remote Wakeup

Another PCI Express device can initiate a Device-Remote Wakeup, by asserting WAKE# to the USB 3382. After the USB 3382 completes the wakeup process, the 8051 writes to the **USBSTAT** register *Generate Resume* bit (USB Controller, offset 90h[5]). This transmits a Device Remote Wake Resume signal to the USB Host.

14.6.4 Resume Interrupt

When the USB 3382 starts a Device-Remote or Host-Initiated Wakeup, the USB 3382 can also be programmed to generate a Resume interrupt. The **IRQSTAT1** register *Resume Interrupt Status* bit (USB Controller, offset 2Ch[1]) is Set when a resume is detected, and enabled to generate an interrupt when the *Resume Interrupt Enable* bit is Set.

14.7 USB Low-Power Suspend State

The USB 3382 has the option of entering the USB Low-Power Suspend state, when the USB interface enters the Suspended state, with most of its circuitry powered down. In the USB Low-Power Suspend state, a small portion of the USB 3382 that detects the Host wakeup and device remote wakeup is operational, which ensures that the suspend current from the USB_VBUS input is as low as 2.5 mA. This circuit mostly operates on the USB cable's VBUS pin. Main power to the USB 3382 can be controlled by LANE_GOOD0# (GPIO0) output, which in turn controls a power switch MOSFET. In this state, LANE_GOOD1# (GPIO1) is operated in the always ON power domain, and can be used to control the PCI Express Adapter (endpoint) device's clock reset and power.

Parameters Set by the USB Host (*such as* device address, configuration, and U1/U2_enable) are held in this power state. After USB Host or Device Resume is detected, the USB 3382 powers itself, using an external power switch that is controlled by LANE_GOOD0#/GPIO0. A full re-programming of the register set, through serial EEPROM/USB Host/8051 or PCI Express Host, is required, to resume traffic after coming out of the USB Low-Power Suspend state.

In Adapter mode, where the USB 3382 resides in a PCI Express system, the suspend current can be provided from the PCI Express Vaux supply. In such systems, the USB 3382 has the capability of waking up either from USB Host Resume, or PCI Express WAKE#, or beacon events from PCI Express Adapter (endpoint) devices. In such cases, the PCI Express Host can wakeup the USB Host, after the USB 3382 is powered up and the PCI Express interface enters the L0 Link PM state.

Table 14-5 lists supported and non-supported features, and briefly describes the register bits/fields used for configuration or activation. For further details, refer to the referenced register bits/fields, in Chapter 16, "USB Configuration Registers."

Table 14-5. USB Power Management Support Registers

Register		Description	Supported	
Offset	Bit(s)		Yes	No
		PCIIRQENB1 (USB Controller)		
14h	1	**Resume PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when the USB 3382 resumes from the Suspended state	✔	
14h	2	**Suspend Request Change PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when a change in the Suspend Request Interrupt state is detected	✔	
14h	27	**Power State Change PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when the **PCI Power Management Status and Control** register *Power State* field (All Ports and USB Controller, offset 44h[1:0]) changes	✔	
14h	30	**PCI Express Endpoint Power Management PCI Express Interrupt Enable** ***Valid only in Adapter mode.*** PCI Express Adapter (endpoint) enable for Power Management PCI Express interrupts.	✔	

Table 14-5. USB Power Management Support Registers *(Cont.)*

Register Offset	Register Bit(s)	Description	Supported Yes	Supported No
		CPUIRQENB1 (USB Controller)		
1Ch	1	**Resume 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when the USB 3382 resumes from the Suspended state	✔	
	2	**Suspend Request Change 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when a change in the Suspend Request Interrupt state is detected	✔	
	27	**Power State Change 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when the **PCI Power Management Status and Control** register *Power State* field (All Ports and USB Controller, offset 44h[1:0]) changes	✔	
	30	**PCI Express Endpoint Power Management 8051 Interrupt Enable** ***Valid only in Enhanced Adapter mode.*** 1 = Enables ability to generate an 8051 interrupt to the PCI Express Adapter (endpoint)	✔	
		USBIRQENB1 (USB Controller)		
24h	1	**Resume USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when the USB 3382 resumes from the Suspended state	✔	
	2	**Suspend Request Change USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when a change in the Suspend Request Interrupt state is detected	✔	
	27	**Power State Change USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when the **PCI Power Management Status and Control** register *Power State* field (All Ports and USB Controller, offset 44h[1:0]) changes	✔	
	30	**PCI Express Endpoint Power Management USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt to the PCI Express Adapter (endpoint)	✔	
		IRQSTAT1 (USB Controller)		
2Ch	1	**Resume Interrupt Status** 1 = Indicates that the USB 3382 resumed from the Suspended state	✔	
	2	**Resume Interrupt Status** 1 = Suspend Request Interrupt state (bit 3) changed	✔	
	3	**Resume Interrupt Status** 1 = USB 3382 detected a USB Suspend Request from the Host	✔	
	27	**Power State Change Interrupt Status** 1 = **PCI Power Management Status and Control** register *Power State* field (All Ports and USB Controller, offset 44h[1:0]) changed	✔	
	30	**PCI Express Endpoint Power Management Interrupt Status** PCI Express Adapter (endpoint) Power Management PCI interrupt.	✔	

Table 14-5. USB Power Management Support Registers *(Cont.)*

Register		Description	Supported	
Offset	Bit(s)		Yes	No
		USBCTL (USB Controller)		
8Ch	1	**Remote Wakeup Enable** 1 = Enables the Device Remote Wakeup feature	✔	
	2	**PCI Express Wakeup Enable** 1 = Enables the PCI Express WAKE# pin or beacon, to wake up the USB 3382	✔	
	5	**Remote Wakeup Support** Indicates whether the USB 3382 supports Device Remote Wakeup.	✔	
	7	**Immediately Suspend** 0 = IRQSTAT1 register Suspend Request Interrupt Status bit (USB Controller, offset 2Ch[3]) must be written to initiate the Suspend sequence 1 = Allows the USB 3382 to automatically enter the Suspend state when the USB is idle for 3 ms. Automatically Set if Root Complex mode is selected and a valid serial EEPROM is not detected at reset time.	✔	
	11	**USB Root Port Wakeup Enable** 0 = Wakeup condition is not detected 1 = Root Port Wakeup condition is detected when activity is detected on the USB line interface	✔	
		USBSTAT (USB Controller)		
90h	5	**Generate Resume** Writing 1 initiates a Resume sequence to the Host, if Device Remote Wakeup is enabled (USBCTL register Remote Wakeup Enable bit (USB Controller, offset 8Ch[1]) is Set).	✔	
	9	**Suspend Status** 1 = Indicates that the USB 3382 was previously in the Suspend state	✔	
		USBPM Control (USB Controller)		
6C0h	0	**USB Mode PME to ACK Send Enable** 1 = In Adapter mode, enables PME TO ACK Message generation in response to a received PME_Turn_Off Message	✔	
	1	**PME to ACK in Suspend Only** 1 = In Adapter mode, enables PME to ACK generation only when the USB interface is in the Suspend state	✔	
	2	**USB2 L1 STATIN Pending Device Remote Wakeup Enable** 1 = *USB r2.0* L1 Link PM state remote wakeup due to a pending STATIN Dedicated Endpoint interrupt. The USB 3382 returns a NYET handshake for L1 Requests, if a STATIN interrupt is pending.	✔	
	3	**USB2 L1 USB IN FIFO Packet Pending Device Remote Wakeup Enable** 1 = Enables *USB r2.0* L1 Link PM state remote wakeup due to a pending USB IN FIFO Data packet. The USB 3382 returns a NYET handshake for L1 Requests, if the USB IN FIFOs are not empty.	✔	
	4	**USB Suspend STATIN Pending Device Remote Wakeup Enable** 1 = Enables USB Suspend Device Remote Wakeup due to a pending STATIN interrupt	✔	
	5	**USB Suspend IN FIFO Packet Pending Device Remote Wakeup Enable** 1 = Enables USB Suspend Device Remote Wakeup due to a pending USB IN FIFO Data packet	✔	

Chapter 15 PCI Configuration Registers

15.1 Introduction

This chapter defines the USB 3382 PCI Configuration registers. Each USB 3382 Port and the SuperSpeed USB has its own PCI Configuration register space. The register mapping is similar for each Port/the SuperSpeed USB. (Refer to Table 15-1.) This chapter also presents the USB 3382 PCI Configuration registers and the order in which they appear in the register map. Register descriptions, when applicable, include details regarding their use and meaning in Port 0, downstream Ports, and the SuperSpeed USB. (Refer to Table 15-4.)

All USB 3382 registers can be accessed by Configuration or Memory Requests.

Other registers are defined in Section 16, "USB Configuration Registers."

For further details regarding register names and descriptions, refer to the following specifications:

- *PCI r3.0*
- *PCI Power Mgmt. r1.2*
- *P-to-P Bridge r1.1*
- *PCI Express Base r2.1*

15.2 PCI Configuration Register Map

Table 15-1 defines the PCI Configuration register mapping.

Table 15-1. PCI Configuration Register Map

31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
PCI Express Interface (Ports 0, 1, and 2) – PCI-Compatible Type 1 Configuration Header Registers (Offsets 00h – 3Ch) USB Controller – PCI-Compatible Type 0 Configuration Header Registers (Offsets 00h – 3Ch)			Capability Pointer (40h)	00h … 34h … 3Ch
		Next Capability Pointer (48h)	Capability ID (01h)	40h
PCI Power Management Capability Registers (Offsets 40h – 44h)				44h
		Next Capability Pointer (68h)	Capability ID (05h)	48h
MSI Capability Registers (Offsets 48h – 64h)				… 64h
		Next Capability Pointer (00h)	Capability ID (10h)	68h
PCI Express Capability Registers (Offsets 68h – A0h)				… A0h
Reserved			A4h –	FCh
Next Capability Offset (FB4h)	1h	PCI Express Extended Capability ID (0003h)		100h
Device Serial Number Extended Capability Registers (Offsets 100h – 134h)				… 134h
Next Capability Offset (148h)	1h	PCI Express Extended Capability ID (0004h)		138h
Power Budget Extended Capability Registers (Offsets 138h – 144h)				… 144h
Next Capability Offset (950h)	1h	PCI Express Extended Capability ID (0002h)		148h
Virtual Channel Extended Capability Registers (Offsets 148h – 1BCh)				… 1BCh
Device-Specific Registers (Offsets 1C0h – 444h)				1C0h … 51Ch
Reserved			520h –	52Ch

Table 15-1. PCI Configuration Register Map *(Cont.)*

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16		15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Device-Specific Registers (Offsets 530h – B88h)			530h …
Next Capability Offset 2 (000h)	1h	PCI Express Extended Capability ID 2 (000Bh)	950h
Device-Specific Registers (Offsets 530h – B88h)			… B88h
Factory Test Only/Reserved			B8Ch – FB0h
Next Capability Offset (138h)	1h	PCI Express Extended Capability ID (0001h)	FB4h
Advanced Error Reporting Extended Capability Registers (Offsets FB4h – FE8h)			… FE8h
Reserved			FECh – FFCh

15.3 Register Configuration and Map

The USB 3382 PCI Configuration registers are configured similarly – not all the same. Port 0 includes more Device-Specific registers than the Port 1 and the SuperSpeed USB. Port 0 also contains registers that are used to set up and control the USB 3382, as well as a serial EEPROM interface, I^{2}C Slave interface, and SMBus Slave interface logic and control.

Table 15-2 defines the register configuration and map.

Table 15-2. Register Configuration and Map

Register Types	Port 0	Ports 1 and 2	USB Controller
PCI Express Interface (Ports 0, 1, and 2) – PCI-Compatible Type 1 Configuration Header Registers (Offsets 00h – 3Ch)	00h – 3Ch	00h – 3Ch	
USB Controller – PCI-Compatible Type 0 Configuration Header Registers (Offsets 00h – 3Ch)			00h – 3Ch
PCI Power Management Capability Registers (Offsets 40h – 44h)	40h – 44h	40h – 44h	40h – 44h
MSI Capability Registers (Offsets 48h – 64h)	48h – 64h	48h – 64h	48h – 64h
PCI Express Capability Registers (Offsets 68h – A0h)	68h – A0h	68h – 80h 8Ch – A0h	68h – A0h
Device Serial Number Extended Capability Registers (Offsets 100h – 134h)	100h – 134h	100h – 134h	100h – 134h
Power Budget Extended Capability Registers (Offsets 138h – 144h)	138h – 144h		
Virtual Channel Extended Capability Registers (Offsets 148h – 1BCh)	148h – 1BCh	148h – 1BCh	148h – 1BCh
Device-Specific Registers (Offsets 1C0h – 444h)			
Device-Specific Registers – Error Checking and Debug (Offsets 1C0h – 1FCh)	1C0h – 1FCh	1E0h – 1ECh, 1F8h, 1FCh	1E0h – 1ECh, 1F8h, 1FCh
Device-Specific Registers – Physical Layer (Offsets 200h – 25Ch)	200h – 25Ch		
Device-Specific Registers – Serial EEPROM (Offsets 260h – 26Ch)	260h – 26Ch		
Device-Specific Registers – I^{2}C and SMBus Slave Interfaces (Offsets 290h – 2C4h)	290h – 2C4h		
Device-Specific Registers (Offsets 530h – B88h)			
Device-Specific Registers – Port Configuration (Offset 574h)	574h		
Device-Specific Registers – Negotiated Link Width (Offsets 660h – 67Ch)	660h – 67Ch		
Device-Specific Registers – Vendor-Specific Extended Capability 2 (Offsets 950h – 95Ch)	950h – 95Ch	950h – 95Ch	950h – 95Ch
Device-Specific Registers – Ingress Credit Handler Control and Status (Offsets 9F0h – 9FCh)	9F0h – 9FCh		

Table 15-2. Register Configuration and Map *(Cont.)*

Register Types	Port 0	Ports 1 and 2	USB Controller
Device-Specific Registers – Ingress Credit Handler Threshold (Offsets A00h – A2Ch)	A00h – A2Ch		
Device-Specific Registers – Physical Layer (Offsets B80h – B88h)	B80h – B88h		
Advanced Error Reporting Extended Capability Registers (Offsets FB4h – FE8h)	FB4h – FDCh	FB4h – FDCh	FB4h – FE8h

Table 15-3 lists registers that are generally individual registers that support all Ports and the SuperSpeed USB (changing the register value in one Port/SuperSpeed USB changes the same register in the other Ports/SuperSpeed USB).

Table 15-3. Singular Registers Shared by All Ports and SuperSpeed USB

Offset	Register	Comment
00h	Vendor ID and Device ID	
08h	PCI Class Code and Revision ID	
34h	Capability Pointer	
100h	Device Serial Number Extended Capability Header	
104h	Serial Number (Lower DW)	
108h	Serial Number (Upper DW)	
950h	Vendor-Specific Extended Capability 2	
954h	Vendor-Specific Header 2	
958h	Hardwired Vendor ID and Hardwired Device ID	
95Ch	Hardwired Revision ID	

15.4 Register Access

Each Port and the SuperSpeed USB implement a 4-KB Configuration Space. The lower 256 bytes (offsets 00h through FFh) comprise the PCI-compatible Configuration Space, and the upper 960 Dwords (offsets 100h through FFFh) comprise the PCI Express Extended Configuration Space. The USB 3382 supports six mechanisms for accessing the registers:

- PCI r3.0-Compatible Configuration Mechanism
- PCI Express Enhanced Configuration Access Mechanism
- Device-Specific Memory-Mapped Configuration Mechanism
- I^2C Slave Interface (refer to Section 12.2, "I^2C Slave Interface")
- SMBus Slave Interface (refer to Section 12.3, "SMBus Slave Interface")
- Serial Peripheral Interface (SPI) Bus (refer to Chapter 5, "Serial EEPROM Controller")

The sideband register access mechanisms (serial EEPROM, I^2C, and/or SMBus) can modify Read-Only (RO) register values.

Each Port captures the Bus Number and Device Number on every Type 0 Configuration Write, as required by the *PCI r3.0*. Therefore, following a Fundamental Reset, software must initially perform a Configuration Write to each Port (using either the PCI r3.0-Compatible Configuration Mechanism or PCI Express Enhanced Configuration Access Mechanism), to allow each Port to capture its designated Bus Number and Device Number. The initial access to each Port, *for example*, could be a Configuration Write Request to a RO register, *such as* the **Device ID** / **Vendor ID** register (All Ports, offset 00h).

15.4.1 *PCI r3.0*-Compatible Configuration Mechanism

The *PCI r3.0*-Compatible Configuration mechanism provides standard access to the USB 3382 Ports' and the SuperSpeed USB's first 256 bytes (the bytes at offsets 00h through FFh) of the PCI Express Configuration Space. The mechanism uses PCI Type 0 and Type 1 Configuration transactions to access the USB 3382 Configuration registers. Each Port/SuperSpeed USB can convert a Type 1 Configuration Request (destined to a downstream Port or device) to a Type 0 Configuration Request (targeting the next downstream Port or device), as described below. The USB Controller responds only to Type 0 accesses. In Legacy Adapter mode, the USB 3382 responds only to Type 0 Requests on the PCI Express interface. In Enhanced Adapter mode, Type 1 Requests received by Port 0 are automatically converted to Type 0 Requests.

The USB 3382 decodes all Type 1 Configuration accesses received on Port 0, when any of the following conditions exist:

- If the Bus Number in the Configuration access is not within Port 0's Secondary Bus Number and Subordinate Bus Number range, Port 0 responds with an Unsupported Request (UR).
- Specified Bus Number in the Configuration access is the USB 3382 internal virtual PCI Bus Number, the USB 3382 automatically converts the Type 1 Configuration access into the appropriate Type 0 Configuration access for the specified device.
  - If the specified device corresponds to the PCI-to-PCI bridge in one of the USB 3382 downstream Ports, the USB 3382 processes the Read or Write Request to the specified downstream Port register specified in the original Type 1 Configuration access.
  - If the specified Device Number does not correspond to any of the USB 3382 downstream Port Device Numbers, the USB 3382 responds with a UR.

- If the specified Bus Number in the Type 1 Configuration access is not the USB 3382 internal virtual PCI Bus Number, but is the Bus Number of one of the USB 3382 downstream Port secondary/subordinate buses, the USB 3382 passes the Configuration access on to the PCI Express Link attached to that USB 3382 downstream Port.
  - If the specified Bus Number is the downstream Port Secondary Bus Number, and the specified Device Number is 0, the USB 3382 converts the Type 1 Configuration access to a Type 0 Configuration access before passing it on.
  - If the specified Device Number is not 0, the downstream Port drops the Transaction Layer Packet (TLP) and generates a UR.
- If the specified Bus Number is not the downstream Port Secondary Bus Number, the USB 3382 passes along the Type 1 Configuration access, without change.

Because the mechanism is limited to the first 256 bytes of the PCI Express Configuration Space of the USB 3382 Ports and SuperSpeed USB, the PCI Express Enhanced Configuration Access Mechanism or Device-Specific Memory-Mapped Configuration Mechanism must be used to access beyond Byte FFh. The PCI Express Enhanced Configuration Access mechanism can access the registers in the PCI-compatible region, as well as those in the PCI Express Extended Configuration Space.

15.4.2 PCI Express Enhanced Configuration Access Mechanism

The PCI Express Enhanced Configuration Access mechanism is implemented on all PCI Express PCs and on systems that do not implement a processor-specific firmware interface to the Configuration Space. The mechanism provides a Memory-Mapped Address space in the Root Complex, through which the Root Complex translates a Memory access into one or more Configuration Requests. Device drivers normally use an application programming interface (API) provided by the Operating System (OS), to use this mechanism.

The mechanism can be used to access all USB 3382 registers.

15.4.3 Device-Specific Memory-Mapped Configuration Mechanism

The Device-Specific Memory-Mapped Configuration mechanism provides a method to access the Configuration registers of all Ports and the USB Controller within a single 128-KB Memory map, as listed in Table 15-4. The registers are contained within a 4-KB range. To use this mechanism in Enhanced Adapter mode, program the Port 0 Type 1 Configuration Space **Base Address 0** and **Base Address 1** registers (**BAR0** and **BAR1**, offsets 10h and 14h, respectively), which are typically enumerated at boot time by BIOS and/or the OS software. After the Port 0 BARs are enumerated, Port 0 registers can be accessed with Memory Reads from and Writes to the first 4 KB (0000h to 0FFFh), Port 1 registers can be accessed with Memory Reads from and Writes to the second 4 KB (1000h to 1FFFh). Port 2 registers can be accessed with Memory Reads from and Writes to the third 4 KB (2000h to 2FFFh). (Refer to Table 15-4.) Within each of these 4-KB windows, individual registers are located at the DWord offsets indicated in Table 15-1.

Port 0 **BAR0** and **BAR1** are typically enumerated at boot time, by BIOS and/or the OS software.

USB Controller registers, for both Legacy and Enhanced modes, are accessible from the PCI Express interface, using Memory Read/Write accesses that are offset from the USB Controller Type 0 Header **BAR0/1** registers (**BAR0** and **BAR1**, offsets 10h and 14h, respectively), as listed in Table 15-5.

This mechanism follows the *PCI Express Base r2.1* Configuration Request Routing rules, which do not allow the propagation of Configuration Requests from downstream-to-upstream, nor peer-to-peer. By default, if a downstream Port receives a Memory Request from a downstream device targeting the USB 3382 Configuration registers, the Port or SuperSpeed USB:

- Responds to a Memory Read Request with a UR
- By default, silently discards a Memory Write Request (in compliance with the *PCI Express Base r2.1*)

In Memory Requests that target USB 3382 registers, the Payload Length indicated within the Memory Request Header must be 1 DWord. Lengths greater than 1 DWord result in a Completer Abort error.

Table 15-4. Port Register Offsets from Port 0 BAR0/1 Base Address (Enhanced Adapter Mode)

Port Number	Register Offset from Port 0 BAR0/1	Location Range
0 (upstream)	0000h to 0FFFh	0 to 4 KB
1 (downstream)	1000h to 1FFFh	4 to 8 KB
2(downstream; virtual Port connecting to USB Controller endpoint)	2000h to 2FFFh	8 to 12 KB

Table 15-5. USB Controller Register Offsets from USB Controller BAR0/1 Base Address (Legacy and Enhanced Adapter Modes)

Register Block	Register Offset from USB Controller BAR0/1	Location Range
USB Controller Configuration registers	0000h to 0FFFh	0 to 4 KB
USB Controller PCI Configuration registers (Type 0)	1000h to 1FFFh	4 to 8 KB

15.5 Register Descriptions

The remainder of this chapter details the USB 3382 registers, including:

- Bit/field names
- Description of register functions for each Port and the SuperSpeed USB
- Type (*such as* RW or HwInit; refer to Table 15-6 for Type descriptions)
- Whether the power-on/reset value can be modified, by way of the USB 3382 serial EEPROM and/or I^{2}C/SMBus Initialization feature
- Default power-on/reset value

Table 15-6. Register Types, Grouped by User Accessibility

Type	Description
HwInit	**Hardware-Initialized** Refers to the USB 3382 Hardware-Initialization mechanism or USB 3382 Serial EEPROM or I^{2}C register Initialization features. RO after initialization and can only be reset with a Fundamental Reset. HwInit register bits are not modified by a Soft Reset.
RO	**Read-Only** Read-Only and cannot be altered by software. Permitted to be initialized by the USB 3382 Hardware-Initialization mechanism or USB 3382 serial EEPROM and/or I^{2}C register Initialization features.
ROS	**Read-Only, Sticky** Same as RO, except that bits are neither initialized nor modified by a Soft Reset.
RsvdP	***Reserved* and Preserved** ***Reserved*** for future RW implementations. Registers are RO and must return a value of 0 when read. Software must preserve the value read for Writes to bits.
RsvdZ	***Reserved* and Zero** ***Reserved*** for future RW1C implementations. Registers are RO and must return a value of 0 when read. Software must use 0 for Writes to bits.
RW	**Read-Write** Read/Write and permitted to be Set or Cleared by software to the needed state.
RW1C	**Write 1 to Clear Status** Indicates status when read. A status bit Set by the system (to indicate status) is Cleared by writing 1 to that bit. Writing 0 to the bit has no effect.
RW1CS	**Write 1 to Clear, Sticky** Same as RW1C, except that bits are neither initialized nor modified by a Soft Reset.
RWS	**Read-Write, Sticky** Same as RW, except that bits are Set or Cleared by software to the needed state. Bits are neither initialized nor modified by a Soft Reset.
RZ	**Software Read Zero** Software Read always returns a value of 0; however, software is allowed to write this register.
W1RZ	**Write 1 to Clear, Software Read Zero** Write 1 to activate the function. Reads to this bit always return a value of 0.
WO	**Write-Only.**

15.6 PCI-Compatible Type 1 Configuration Header Registers (Offsets 00h – 3Ch)

This section details the PCI-Compatible Type 1 Configuration Header registers. Table 15-7 defines the register map.

Table 15-7. PCI-Compatible Type 1 Configuration Header Register Map

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Device ID		Vendor ID		00h
PCI Status		PCI Command		04h
PCI Class Code			PCI Revision ID	08h
PCI BIST *(Reserved)*	PCI Header Type	Master Latency Timer *(Not Supported)*	Cache Line Size	0Ch
Base Address 0				10h
Base Address 1				14h
Secondary Latency Timer *(Not Supported)*	Subordinate Bus Number	Secondary Bus Number	Primary Bus Number	18h
Secondary Status	***Not Supported/Reserved***	I/O Limit	I/O Base	1Ch
Memory Limit		Memory Base		20h
Prefetchable Memory Limit		Prefetchable Memory Base		24h
Prefetchable Memory Upper Base Address				28h
Prefetchable Memory Upper Limit Address				2Ch
I/O Limit Upper 16 Bits		I/O Base Upper 16 Bits		30h
Reserved			Capability Pointer (40h)	34h
Expansion ROM Base Address ***(Reserved)***				38h
Not Supported/Reserved	Bridge Control	PCI Interrupt Pin	PCI Interrupt Line	3Ch

Register 15-1. 00h PCI Configuration ID (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
15:0	**Vendor ID** Identifies the device manufacturer. Defaults to the PCI-SIG-issued Vendor ID of PLX (10B5h), if not overwritten by serial EEPROM and/or I^2C.	RO	Yes	10B5h
31:16	**Device ID** Identifies the particular device. Defaults to the PLX part number for the USB 3382, if not overwritten by serial EEPROM and/or I^2C.	RO	Yes	3382h

Register 15-2. 04h PCI Command/Status (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
PCI Command				
0	**I/O Access Enable** 0 = USB 3382 ignores I/O Space accesses on the Port’s primary interface 1 = USB 3382 responds to I/O Space accesses on the Port’s primary interface	RW	Yes	0
1	**Memory Access Enable** 0 = USB 3382 ignores Memory Space accesses on the Port’s primary interface 1 = USB 3382 responds to Memory Space accesses on the Port’s primary interface	RW	Yes	0
2	**Bus Master Enable** Controls USB 3382 Memory and I/O Request forwarding upstream. Neither affect Message (including INT*x* Interrupt Messages) forwarding nor Completions traveling upstream or downstream. 0 = USB 3382 handles Memory and I/O Requests received on the Port’s downstream/secondary interface as Unsupported Requests (URs); for Non-Posted Requests, the USB 3382 returns a Completion with UR Completion status. Because MSI Messages are in-band Memory Writes, disables MSI Messages as well. 1 = USB 3382 forwards Memory and I/O Requests upstream.	RW	Yes	0
3	**Special Cycle Enable** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
4	**Memory Write and Invalidate Enable** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
5	**VGA Palette Snoop** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
6	**Parity Error Response Enable** Controls bit 24 (*Master Data Parity Error Detected*).	RW	Yes	0
7	**IDSEL Stepping/Wait Cycle Control** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0

Register 15-2. 04h PCI Command/Status (All Ports) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
8	**SERR# Enable** Controls bit 30 (*Signaled System Error*). 1 = Enables reporting of Fatal and Non-Fatal errors detected by the device to the Root Complex, and, enables primary interface forwarding of ERR_FATAL and ERR_NONFATAL Messages from downstream Ports and devices when the Port's **Bridge Control** register *SERR# Enable* bit (All Ports, offset 3Ch[17]) is Set	RW	Yes	0
9	**Fast Back-to-Back Transactions Enable** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
10	**Interrupt Disable** 0 = Port is enabled to generate INT*x* Interrupt Messages and assert PEX_INTA# output 1 = Port is prevented from generating INT*x* Interrupt Messages and asserting PEX_INTA# output	RW	Yes	0
15:11	***Reserved***	RsvdP	No	0-0h

Register 15-2. 04h PCI Command/Status (All Ports) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
PCI Status				
18:16	***Reserved***	RsvdP	No	000b
19	**Interrupt Status** 0 = No INT*x* Interrupt Message is pending 1 = INT*x* Interrupt Message is pending internally to the Port –or– PEX_INTA# (if enabled) is asserted	RO	No	0
20	**Capability List** Capability function is supported. Set, as required by the *PCI Express Base r2.1*.	RO	Yes	1
21	**66 MHz Capable** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
22	***Reserved***	RsvdP	No	0
23	**Fast Back-to-Back Transactions Capable** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
24	**Master Data Parity Error Detected** If bit 6 (*Parity Error Response Enable*) is Set, the Port Sets this bit when the Port: • Forwards the poisoned Transaction Layer Packet (TLP) Write Request from the secondary to the primary interface, –or– • Receives a Completion marked as poisoned on the primary interface If the *Parity Error Response Enable* bit is Cleared, the USB 3382 never Sets this bit. This error is natively reported by the **Uncorrectable Error Status** register *Poisoned TLP Status* bit (All Ports and USB Controller, offset FB8h[12]), which is mapped to this bit for Conventional PCI backward compatibility.	RW1C	Yes	0
26:25	**DEVSEL# Timing** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	00b
27	**Signaled Target Abort** Port 0 Sets this bit if one of the following conditions exist: • Port 0 receives a Memory Request targeting a USB 3382 register, and the Payload Length (indicated within the Memory Request Header) is greater than 1 DWord • Port 0 receives a Memory Request targeting a USB 3382 register address within a non-existent Port This error is reported by the **Uncorrectable Error Status** register *Completer Abort Status* bit (All Ports and USB Controller, offset FB8h[15]), which is mapped to this bit for Conventional PCI backward compatibility.	RW1C	Yes	0

Register 15-2. 04h PCI Command/Status (All Ports) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
28	**Received Target Abort** ***Reserved***	RsvdP	No	0
29	**Received Master Abort** ***Reserved***	RsvdP	No	0
30	**Signaled System Error** If bit 8 (*SERR# Enable*) is Set, the Port Sets this bit when it transmits or forwards an ERR_FATAL or ERR_NONFATAL Message upstream. This error is natively reported by the **Device Status** register *Fatal Error Detected* and *Non-Fatal Error Detected* bits (All Ports and USB Controller, offset 70h[18:17], respectively), which are mapped to this bit for Conventional PCI backward compatibility.	RW1C	Yes	0
31	**Detected Parity Error** This error is natively reported by the **Uncorrectable Error Status** register *Poisoned TLP Status* bit (All Ports and USB Controller, offset FB8h[12]), which is mapped to this bit for Conventional PCI backward compatibility. 1 = Port received a Poisoned TLP on its primary side, regardless of the bit 6 (*Parity Error Response Enable*) state	RW1C	Yes	0

Register 15-3. 08h PCI Class Code and Revision ID (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
PCI Revision ID				
7:0	**Revision ID** Unless overwritten by the serial EEPROM, returns the Silicon Revision (AAh or ABh), the PLX-assigned Revision ID for this version of the USB 3382. The USB 3382 Serial EEPROM register Initialization capability is used to replace the PLX Revision ID with another Revision ID.	RO	Yes	AAh or ABh
PCI Class Code				**060400h**
15:8	**Register-Level Programming Interface** The USB 3382 Ports support the *P-to-P Bridge r1.1* requirements, but not subtractive decoding, on their upstream interface.	RO	Yes	00h
23:16	**Sub-Class Code** PCI-to-PCI bridge.	RO	Yes	04h
31:24	**Base Class Code** Bridge device.	RO	Yes	06h

Register 15-4. 0Ch Miscellaneous Control (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Cache Line Size				
7:0	**Cache Line Size** System Cache Line Size. Implemented as a RW field for Conventional PCI compatibility purposes and does not impact USB 3382 functionality.	RW	Yes	00h
Master Latency Timer				
15:8	**Master Latency Timer** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	00h
PCI Header Type				
22:16	**Configuration Layout Type** The Port's Configuration Space Header adheres to the Type 1 PCI-to-PCI Bridge Configuration Space layout defined by the *P-to-P Bridge r1.1*.	RO	No	01h
23	**Multi-Function Device** 0 = Single-function device 1 = Indicates multiple (up to eight) functions (logical devices), each containing its own, individually addressable Configuration Space, 256 DWords in size	RO	No	0
PCI BIST				
31:24	**PCI BIST** ***Reserved*** Built-In Self-Test (BIST) Pass or Fail.	RsvdP	No	00h

Register 15-5. 10h Base Address 0 (Port 0)

Bit(s)	Description	Ports	Type	Serial EEPROM and I^2C	Default
0	**Memory Space Indicator** 0 = Base Address register maps the USB 3382 Configuration registers into Memory space ***Note:*** *Port 0 is hardwired to 0.*	Port 0	RO	No	0
	Reserved	Downstream	RsvdP	No	0
2:1	**Memory Map Type** 00b = Base Address register is 32 bits wide and can be mapped anywhere in the 32-bit Memory space 10b = Base Address register is 64 bits wide and can be mapped anywhere in the 64-bit Address space All other encodings are ***reserved***.	Port 0	RO	Yes	00b
	Reserved	Downstream	RsvdP	No	00b
3	**Prefetchable** 0 = Base Address register maps the USB 3382 Configuration registers into Non-Prefetchable Memory space ***Note:*** *Port 0 is hardwired to 0.*	Port 0	RO	Yes	0
	Reserved	Downstream	RsvdP	No	0
13:4	***Reserved***		RsvdP	No	0-0h
31:14	**Base Address 0** Base Address (**BAR0**) for the Device-Specific Memory-Mapped Configuration mechanism.	Port 0	RW	Yes	0-0h
	Reserved	Downstream	RsvdP	No	0-0h

Register 15-6. 14h Base Address 1 (Port 0)

Bit(s)	Description	Ports	Type	Serial EEPROM and I^2C	Default
31:0	**Base Address 1** RO when the **Base Address 0** (**BAR0**) register is not enabled as a 64-bit BAR (*Memory Map Type* field (Port 0, offset 10h[2:1]) is not programmed to 10b).	Port 0	RW	Yes	0000_0000h
	For 64-bit addressing (**BAR0/1**), Base Address 1 (**BAR1**) extends Base Address 0 (**BAR0**) to provide the upper 32 Address bits when the **Base Address 0** register *Memory Map Type* field (Port 0, offset 10h[2:1]) is programmed to 10b.	Downstream	RsvdP	Yes	0000_0000h

Register 15-7. 18h Bus Number (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
7:0	**Primary Bus Number** Primary Bus Number of this PCI-to-PCI bridge. Records the Bus Number of the PCI Bus segment to which the primary interface of this Port is connected. Set by Configuration software.	RW	Yes	00h
15:8	**Secondary Bus Number** Secondary Bus Number of this PCI-to-PCI bridge. Records the Bus Number of the PCI Bus segment that is the secondary interface of this Port. Set by Configuration software.	RW	Yes	00h
23:16	**Subordinate Bus Number** Subordinate Bus Number of this PCI-to-PCI bridge. Records the Bus Number of the highest numbered PCI Bus segment that is subordinate to this Port. Set by Configuration software.	RW	Yes	00h
31:24	**Secondary Latency Timer** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	00h

Register 15-8. 1Ch Secondary Status, I/O Limit, and I/O Base (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
*Note: If ISA Addressing mode is enabled (**PCI Command** register I/O Access Enable bit (All Ports, offset 04h[0]) is Set), the Port forwards I/O transactions from its primary interface to its secondary interface (downstream) if an I/O address is within the range defined by the **I/O Base** and **I/O Limit** registers when the Base is less than or equal to the Limit.* *Conversely, the Port forwards I/O transactions from its secondary interface to its primary interface (upstream) if an I/O address is outside this Address range. If the Port does not implement an I/O Address range, the Port forwards all I/O transactions on its secondary interface upstream, to its primary interface.*				
I/O Base				
3:0	**I/O Base Addressing Capability** 1h = 32-bit I/O Address decoding is supported All other encodings are ***reserved***.	RO	Yes	1h
7:4	**I/O_BAR[15:12]** I/O Base Address[15:12]. The Ports use their **I/O Base** and **I/O Limit** registers to determine the address range of I/O transactions to forward from one interface to the other. I/O Base Address[15:12] bits specify the Port's I/O Base Address[15:12]. The USB 3382 assumes I/O Base Address[11:0]=000h. For 16-bit I/O addressing, the USB 3382 assumes Address[31:16]=0000h. For 32-bit addressing, the USB 3382 decodes Address[31:0], and uses the **I/O Upper Base and Limit Address** register *I/O Base Upper 16 Bits* and *I/O Limit Upper 16 Bits* fields (All Ports, offset 30h[15:0 and 31:16], respectively).	RW	Yes	Fh
I/O Limit				
11:8	**I/O Limit Addressing Capability** 1h = 32-bit I/O Address decoding is supported All other encodings are ***reserved***.	RO	Yes	1h
15:12	**I/O_Limit[15:12]** I/O Limit Address[15:12]. The Ports use their **I/O Base** and **I/O Limit** registers to determine the Address range of I/O transactions to forward from one interface to the other. The I/O Limit Address[15:12] bits specify the Port's I/O Limit Address[15:12]. The USB 3382 assumes Address bits [11:0] of the I/O Limit Address are FFFh. For 16-bit I/O addressing, the USB 3382 decodes Address bits [15:0] and assumes Address bits [31:16] of the I/O Limit Address are 0000h. For 32-bit addressing, the USB 3382 decodes Address bits [31:0], and uses the **I/O Upper Base and Limit Address** register *I/O Base Upper 16 Bits* and *I/O Limit Upper 16 Bits* fields (All Ports, offset 30h[15:0 and 31:16], respectively). If the I/O Limit Address is less than the I/O Base Address, the USB 3382 does not forward I/O transactions from the Port's primary/upstream bus to its secondary/downstream bus. However, the USB 3382 forwards all I/O transactions from the secondary bus of the Port to its primary bus.	RW	Yes	0h

Register 15-8. 1Ch Secondary Status, I/O Limit, and I/O Base (All Ports) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
	Secondary Status			
20:16	***Reserved***	RsvdP	No	0-0h
21	**66 MHz Capable** ***Not supported*** 0 = Not enabled, because PCI Express does ***not support*** 66 MHz	RsvdP	No	0
22	***Reserved***	RsvdP	No	0
23	**Fast Back-to-Back Transactions Capable** ***Reserved*** Not enabled, because PCI Express does ***not support*** this function.	RsvdP	No	0
24	**Master Data Parity Error** If the **Bridge Control** register *Parity Error Response Enable* bit (All Ports, offset 3Ch[16]) is Set, the Port Sets this bit when transmitting or receiving a TLP on its downstream side, and when either of the following two conditions occur: • Port receives Completion marked poisoned • Port forwards poisoned TLP Write Request If the *Parity Error Response Enable* bit is Cleared, the USB 3382 never Sets this bit. These errors are reported by the **Uncorrectable Error Status** register *Poisoned TLP Status* bit (All Ports and USB Controller, offset FB8h[12]), and mirrored to this bit for Conventional PCI backward compatibility.	RW1C	Yes	0
26:25	**DEVSEL# Timing** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	00b
27	**Signaled Target Abort** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
28	**Received Target Abort** Cleared, as required by the *PCI Express Base r2.1*, because the USB 3382 never initiates a Request itself.	RsvdP	No	0
29	**Received Master Abort** Cleared, as required by the *PCI Express Base r2.1*, because the USB 3382 never initiates a Request itself.	RsvdP	No	0
30	**Received System Error** 1 = Downstream Port received an ERR_FATAL or ERR_NONFATAL Message on its secondary interface from a downstream device	RW1C	Yes	0
31	**Detected Parity Error** 1 = Downstream Port received a poisoned TLP from a downstream device, regardless of the **Bridge Control** register *Parity Error Response Enable* bit (All Ports, offset 3Ch[16]) state	RW1C	Yes	0

Register 15-9. 20h Memory Base and Limit (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
Note: *The Port forwards Memory transactions from its primary interface to its secondary interface (downstream) if a Memory address is within the range defined by the* ***Memory Base*** *and* ***Memory Limit*** *registers (when the Base is less than or equal to the Limit).* *Conversely, the Port forwards Memory transactions from its secondary interface to its primary interface (upstream) if a Memory address is outside this Address range (provided that the address is not within the range defined by the* ***Prefetchable Memory Base*** *(All Ports, offsets 28h + 24h[15:0]) and* ***Prefetchable Memory Limit*** *(All Ports, offsets 2Ch + 24h[31:16]) registers).*				
Memory Base				
3:0	***Reserved***	RsvdP	No	0h
15:4	**MEM_BAR[31:20]** Memory Base Address[31:20]. Specifies the Port's Non-Prefetchable Memory Base Address[31:20]. The USB 3382 assumes Memory Base Address[19:0]=0_0000h.	RW	Yes	FFFh
Memory Limit				
19:16	***Reserved***	RsvdP	No	0h
31:20	**MEM_Limit[31:20]** Memory Limit Address[31:20]. Specifies the Port's Non-Prefetchable Memory Limit Address[31:20]. The USB 3382 assumes Memory Limit Address[19:0]=F_FFFFh.	RW	Yes	000h

Register 15-10. 24h Prefetchable Memory Base and Limit (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
Note: The Port forwards Memory transactions from its primary interface to its secondary interface (downstream) if a Memory address is within the range defined by the ***Prefetchable Memory Base*** (All Ports, offsets 28h + 24h[15:0]) and ***Prefetchable Memory Limit*** (All Ports, offsets 2Ch + 24h[31:16]) registers (when the Base is less than or equal to the Limit). *Conversely, the Port forwards Memory transactions from its secondary interface to its primary interface (upstream) if a Memory address is outside this Address range (provided that the address is not within the range defined by the* ***Memory Base*** *and* ***Memory Limit*** *registers (All Ports, offset 20h)).*				
Prefetchable Memory Base				
0	**Prefetchable Memory Base Capability** 0 = Port supports 32-bit Prefetchable Memory Addressing 1 = Port defaults to 64-bit Prefetchable Memory Addressing support, as required by the *PCI Express Base r2.1* ***Note:*** *If the application needs 32-bit only Prefetchable space, the serial EEPROM and/or I²C must Clear both this bit and bit 16 (Prefetchable Memory Limit Capability).*	RO	Yes	1
3:1	***Reserved***	RsvdP	No	000b
15:4	**PMEM_BAR[31:20]** Prefetchable Memory Base Address[31:20]. Specifies the Port's Prefetchable Memory Base Address[31:20]. The USB 3382 assumes Prefetchable Memory Base Address[19:0]=0_0000h.	RW	Yes	FFFh
Prefetchable Memory Limit				
16	**Prefetchable Memory Limit Capability** 0 = Port supports 32-bit Prefetchable Memory Addressing 1 = Port defaults to 64-bit Prefetchable Memory Addressing support, as required by the *PCI Express Base r2.1*	RO	Yes	1
19:17	***Reserved***	RsvdP	No	000b
31:20	**PMEM_Limit[31:20]** Prefetchable Memory Limit Address[31:20]. Specifies the Port's Prefetchable Memory Limit Address[31:20]. The USB 3382 assumes Prefetchable Memory Limit Address[19:0]=F_FFFFh.	RW	Yes	000h

Register 15-11. 28h Prefetchable Memory Upper Base Address (All Ports)

Bit(s)	Description		Type	Serial EEPROM and I²C	Default
31:0	**PBUP[63:32]** Prefetchable Memory Base Address[63:32]. The USB 3382 uses this register for Prefetchable Memory Upper Base Address[63:32]. When the **Prefetchable Memory Base** register *Prefetchable Memory Base Capability* field indicates 32-bit addressing, this register is RO and returns a value of 0000_0000h.	Offset 24h[0]=1	RW	Yes	0000_0000h
		Offset 24h[0]=0	RO	No	0000_0000h

Register 15-12. 2Ch Prefetchable Memory Upper Limit Address (All Ports)

Bit(s)	Description		Type	Serial EEPROM and I²C	Default
31:0	**PLIMUP[63:32]** Prefetchable Memory Limit Address[63:32]. The USB 3382 uses this register for Prefetchable Memory Upper Limit Address[63:32]. When the **Prefetchable Memory Limit** register *Prefetchable Memory Limit Capability* field indicates 32-bit addressing, this register is RO and returns a value of 0000_0000h. ***Note:*** *The serial EEPROM must not write a non-zero value into this register when the RO attribute is Set for this register.*	Offset 24h[16]=1	RW	Yes	0000_0000h
		Offset 24h[16]=0	RO	No	0000_0000h

Register 15-13. 30h I/O Upper Base and Limit Address (All Ports)

Bit(s)	Description		Type	Serial EEPROM and I2C	Default
15:0	**I/O Base Upper 16 Bits** The USB 3382 uses this register for I/O Base Address[31:16]. When the **I/O Base** register *I/O Base Addressing Capability* field indicates 16-bit addressing, this register is RO and returns a value of 0000h.	Offset 1Ch[3:0]=1h	RW	Yes	0000h
		Offset 1Ch[3:0]=0h	RO	No	0000h
31:16	**I/O Limit Upper 16 Bits** The USB 3382 uses this register for I/O Limit Address[31:16]. When the **I/O Limit** register *I/O Limit Addressing Capability* field indicates 16-bit addressing, this register is RO and returns a value of 0000h. ***Note:*** *The serial EEPROM must not write a non-zero value into this register when the RO attribute is Set for this register.*	Offset 1Ch[11:8]=1h	RW	Yes	0000h
		Offset 1Ch[11:8]=0h	RO	No	0000h

Register 15-14. 34h Capability Pointer (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I2C	Default
7:0	**Capability Pointer** Default 40h points to the **PCI Power Management Capability** structure.	RO	Yes	40h
31:8	***Reserved***	RsvdP	No	0000_00h

Register 15-15. 38h Expansion ROM Base Address (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I2C	Default
31:0	**Expansion ROM Base Address** ***Reserved***	RsvdP	No	0000_0000h

Register 15-16. 3Ch Bridge Control and PCI Interrupt Signal (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
PCI Interrupt Line				
7:0	**Interrupt Line Routing Value** The USB 3382 does ***not*** use this register; however, the register is included for operating system and device driver use.	RW	Yes	00h
PCI Interrupt Pin				
15:8	**Interrupt Pin** Identifies the Conventional PCI Interrupt Message(s) that the device (or device function) uses. Only value 00h or 01h is allowed in the USB 3382. 00h = Indicates that the device does not use Conventional PCI Interrupt Message(s) 01h, 02h, 03h, and 04h = Maps to Conventional PCI Interrupt Messages for INTA#, INTB#, INTC#, and INTD#, respectively	RO	Yes	01h
Bridge Control				
16	**Parity Error Response Enable** Controls the response to Poisoned TLPs. 0 = Disables the **Secondary Status** register *Master Data Parity Error* bit (All Ports, offset 1Ch[24]) 1 = Enables the **Secondary Status** register *Master Data Parity Error* bit (All Ports, offset 1Ch[24])	RW	Yes	0
17	**SERR# Enable** Controls forwarding of ERR_COR, ERR_FATAL, and ERR_NONFATAL from the secondary interface to the primary interface. When Set, and the **PCI Command** register *SERR# Enable* bit (All Ports, offset 04h[8]) is Set, enables the **PCI Status** register *Signaled System Error* bit (All Ports, offset 04h[30]).	RW	Yes	0
18	**ISA Enable** Modifies the USB 3382's response to Conventional PCI ISA I/O addresses enabled by the Port's **I/O Base** and **I/O Limit** registers (All Ports, offset 1Ch[7:0 and 15:8], respectively) and located in the first 64 KB of the PCI I/O Address space (0000_0000h to 0000_FFFFh). 0 = Port's **I/O Base** and **I/O Limit** registers, and **I/O Base Upper 16 Bits** and **I/O Limit Upper 16 Bits** registers (All Ports, offset 30h[15:0 and 31:16], respectively), define the Port's I/O Address range 1 = If the Port's I/O Address range (defined by the Port's **I/O Base**, **I/O Limit**, **I/O Base Upper 16 Bits**, **I/O Limit Upper 16 Bits** registers) includes I/O addresses below 64 KB, the upper 768 I/O addresses within each 1-KB block below 64 KB are excluded from the Port's I/O Address range	RW	Yes	0

Register 15-16. 3Ch Bridge Control and PCI Interrupt Signal (All Ports) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
19	**VGA Enable** Modifies the bridge response to VGA-compatible addresses. When Set, the bridge positively decodes and forwards the following addresses on the primary interface to the secondary interface (and, conversely, blocks forwarding of these addresses from the secondary interface to the primary interface): • Memory addresses within the range 000A_0000h to 000B_FFFFh • I/O addresses in the first 64 KB of the I/O Address space (AD[31:16] is 0000h), where AD[9:0] is within the ranges 3B0h to 3BBh and 3C0h to 3DFh (inclusive of ISA address aliases – AD[15:10] is not decoded) Additionally, when Set, forwarding of these addresses is independent of the: • Memory and I/O Address ranges defined by the bridge **I/O Base**, **I/O Limit**, **Memory Base**, **Memory Limit**, **Prefetchable Memory Base**, and **Prefetchable Memory Limit** registers • Bit 18 (*ISA Enable*) Setting VGA address forwarding is qualified by the **PCI Command** register *Memory Access Enable* and *I/O Access Enable* bits (All Ports, offset 04h[1:0], respectively). The default state of this bit after reset must be 0. 0 = Do not forward VGA-compatible Memory and I/O addresses from the primary to the secondary interface (addresses defined above) unless they are enabled for forwarding by the defined Memory and I/O Address ranges 1 = Forward VGA-compatible Memory and I/O addresses (addresses defined above) from the primary interface to the secondary interface (when the *Memory Access Enable* and *I/O Access Enable* bits are Set), independent of the Memory and I/O Address ranges and independent of the *ISA Enable* bit ***Note:* *Conventional PCI VGA support*** *– To avoid potential I/O address conflicts, if the VGA Enable bit is Set in Port 0 and a downstream Port, Set the **PCI Command** register I/O Access Enable bit (All Ports, offset 04h[0]) in the remaining downstream Ports, unless those downstream Ports are configured to use default 32-bit address decoding and their I/O Address range is Set above 1_0000h.*	RW	Yes	0

Register 15-16. 3Ch Bridge Control and PCI Interrupt Signal (All Ports) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
20	**VGA 16-Bit Decode Enable** Used only when bit 19 (*VGA Enable*) is also Set, enabling VGA I/O decoding and forwarding by the bridge. Status after reset is 0. Enables system configuration software to select between 10- and 16-bit I/O address decoding, for VGA I/O register accesses forwarded from the primary interface to the secondary interface. 0 = Execute 10-bit address decodes on VGA I/O accesses 1 = Execute 16-bit address decodes on VGA I/O accesses	RW	Yes	0
21	**Master Abort Mode** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
22	**Secondary Bus Reset** 1 = Causes a Hot Reset on the Port's downstream Link	RW	Yes	0
23	**Fast Back-to-Back Transactions Enable** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
24	**Primary Discard Timer** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
25	**Secondary Discard Timer** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
26	**Discard Timer Status** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
27	**Discard Timer SERR# Enable** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
31:28	***Reserved***	RsvdP	No	0h

15.7 PCI-Compatible Type 0 Configuration Header Registers (Offsets 00h – 3Ch)

This section details the PCI-Compatible Type 0 Configuration Header registers. Table 15-8 defines the register map.

Table 15-8. PCI-Compatible Type 0 Configuration Header Register Map (USB Controller)

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Device ID		Vendor ID		00h
PCI Status		PCI Command		04h
PCI Class Code			PCI Revision ID	08h
PCI BIST ***(Reserved)***	PCI Header Type	Master Latency Timer ***(Not Supported)***	Cache Line Size	0Ch
Base Address 0				10h
Base Address 1				14h
Base Address 2				18h
Base Address 3				1Ch
Base Address 4				20h
Base Address 5				24h
Reserved				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Reserved				30h
Reserved			Capability Pointer (40h)	34h
Reserved				38h
Max_Lat ***(Reserved)***	Min_Gnt ***(Reserved)***	PCI Interrupt Pin	PCI Interrupt Line	3Ch

Register 15-17. 00h PCI Configuration ID (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
15:0	**Vendor ID** Identifies the device manufacturer. Defaults to the PCI-SIG-issued Vendor ID of PLX (10B5h), if not overwritten by serial EEPROM and/or I^2C.	RO	Yes	10B5h
31:16	**Device ID** Identifies the particular device. Defaults to the PLX part number for the USB 3382, if not overwritten by serial EEPROM and/or I^2C.	RO	Yes	3382h

Register 15-18. 04h PCI Command/Status (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
	PCI Command			
0	**I/O Access Enable** 0 = USB 3382 ignores I/O Space accesses on the USB Controller's primary interface 1 = USB 3382 responds to I/O Space accesses on the USB Controller's primary interface	RW	Yes	0
1	**Memory Access Enable** 0 = USB 3382 ignores Memory Space accesses on the USB Controller's primary interface 1 = USB 3382 responds to Memory Space accesses on the USB Controller's primary interface	RW	Yes	0
2	**Bus Master Enable** Controls USB 3382 Memory and I/O Request forwarding upstream. Neither affect Message (including INT*x* Interrupt Messages) forwarding nor Completions traveling upstream or downstream. 0 = USB 3382 handles Memory and I/O Requests received on the USB Controller downstream/secondary interface as Unsupported Requests (URs); for Non-Posted Requests, the USB 3382 returns a Completion with UR Completion status. Because MSI Messages are in-band Memory Writes, disables MSI Messages as well. 1 = USB 3382 forwards Memory and I/O Requests upstream.	RW	Yes	0
3	**Special Cycle Enable** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0

Register 15-18. 04h PCI Command/Status (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
4	**Memory Write and Invalidate Enable** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
5	**VGA Palette Snoop** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
6	**Parity Error Response Enable** Controls bit 24 (*Master Data Parity Error Detected*).	RW	Yes	0
7	**IDSEL Stepping/Wait Cycle Control** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
8	**SERR# Enable** Controls bit 30 (*Signaled System Error*). 1 = Enables reporting of Fatal and Non-Fatal errors detected by the USB Controller to the Root Complex	RW	Yes	0
9	**Fast Back-to-Back Transactions Enable** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
10	**Interrupt Disable** 0 = USB Controller is enabled to generate INT*x* Interrupt Messages and assert PEX_INTA# output 1 = USB Controller is prevented from generating INT*x* Interrupt Messages and asserting PEX_INTA# output	RW	Yes	0
15:11	***Reserved***	RsvdP	No	0-0h

Register 15-18. 04h PCI Command/Status (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
	PCI Status			
18:16	***Reserved***	RsvdP	No	000b
19	**Interrupt Status** 0 = No INT*x* Interrupt Message is pending 1 = INT*x* Interrupt Message is pending internally to the USB Controller –or– PEX_INTA# (if enabled) is asserted	RO	No	0
20	**Capability List** Capability function is supported. Set, as required by the *PCI Express Base r2.1*.	RO	Yes	1
21	**66 MHz Capable** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
22	***Reserved***	RsvdP	No	0
23	**Fast Back-to-Back Transactions Capable** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
24	**Master Data Parity Error Detected** If bit 6 (*Parity Error Response Enable*) is Set, the USB Controller Sets this bit when the USB Controller: • Forwards the poisoned Transaction Layer Packet (TLP) Write Request from the secondary to the primary interface, –or– • Receives a Completion marked as poisoned on the primary interface If the *Parity Error Response Enable* bit is Cleared, the USB 3382 never Sets this bit. This error is natively reported by the **Uncorrectable Error Status** register *Poisoned TLP Status* bit (All Ports and USB Controller, offset FB8h[12]), which is mapped to this bit for Conventional PCI backward compatibility.	RW1C	Yes	0
26:25	**DEVSEL# Timing** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	00b
27	**Signaled Target Abort** The USB Controller Sets this bit if any of the following conditions exist: • USB Controller receives a Completion (from the PCI Express interface) that has a Completion status of Completer Abort (CA), –or– • USB Controller receives a Memory Request targeting a USB 3382 register, and the Payload Length (indicated within the Memory Request Header) is greater than 1 DWord • USB Controller receives a Memory Request targeting a USB 3382 register address within a non-existent Port ***Note:*** *When Set during a forwarded Completion, the* ***Uncorrectable Error Status*** *register Completer Abort Status bit (All Ports and USB Controller, offset FB8h[15]) is not Set.*	RW1C	Yes	0

Register 15-18. 04h PCI Command/Status (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
28	**Received Target Abort**	RW1C	Yes	0
29	**Received Master Abort**	RW1C	Yes	0
30	**Signaled System Error** If bit 8 (*SERR# Enable*) is Set, the USB Controller Sets this bit when it transmits or forwards an ERR_FATAL or ERR_NONFATAL Message upstream. This error is natively reported by the **Device Status** register *Fatal Error Detected* and *Non-Fatal Error Detected* bits (All Ports and USB Controller, offset 70h[18:17], respectively), which are mapped to this bit for Conventional PCI backward compatibility.	RW1C	Yes	0
31	**Detected Parity Error** This error is natively reported by the **Uncorrectable Error Status** register *Poisoned TLP Status* bit (All Ports and USB Controller, offset FB8h[12]), which is mapped to this bit for Conventional PCI backward compatibility. 1 = USB Controller received a Poisoned TLP on its primary side, regardless of the bit 6 (*Parity Error Response Enable*) state	RW1C	Yes	0

Register 15-19. 08h PCI Class Code and Revision ID (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
	PCI Revision ID			
7:0	**Revision ID** Unless overwritten by the serial EEPROM, returns the Silicon Revision (AAh or ABh), the PLX-assigned Revision ID for this version of the USB 3382. The USB 3382 Serial EEPROM register Initialization capability is used to replace the PLX Revision ID with another Revision ID.	RO	Yes	AAh or ABh
	PCI Class Code			**FE030Ch**
15:8	**Register-Level Programming Interface** The USB Controller supports the *P-to-P Bridge r1.1* requirements, but not subtractive decoding, on its upstream interface.	RO	Yes	FEh
23:16	**Sub-Class Code** PCI-to-PCI bridge.	RO	Yes	03h
31:24	**Base Class Code** Bridge device.	RO	Yes	0Ch

Register 15-20. 0Ch Miscellaneous Control (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Cache Line Size				
7:0	**Cache Line Size** System Cache Line Size. Implemented as a RW field for Conventional PCI compatibility purposes and does not impact USB 3382 functionality.	RW	Yes	00h
Master Latency Timer				
15:8	**Master Latency Timer** ***Not supported*** Cleared, as required by the *PCI Express Base r2.1*.	RsvdP	No	00h
PCI Header Type				
22:16	**Configuration Layout Type** The USB Controller Configuration Space Header adheres to the Type 1 PCI-to-PCI Bridge Configuration Space layout defined by the *P-to-P Bridge r1.1*.	RO	No	01h
23	**Multi-Function Device** 0 = Single-function device 1 = Indicates multiple (up to eight) functions (logical devices), each containing its own, individually addressable Configuration Space, 256 DWords in size	RO	No	0
PCI BIST				
31:24	**PCI BIST** ***Reserved*** Built-In Self-Test (BIST) Pass or Fail.	RsvdP	No	00h

Register 15-21. 10h Base Address 0 (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
0	**Memory Space Indicator** 0 = Base Address register maps the USB 3382's USB Configuration registers (refer to Chapter 16, "USB Configuration Registers") into Memory space ***Note:*** *Port 0 is hardwired to 0.*	RO	No	0
	Reserved	RsvdP	No	0
2:1	**Memory Map Type** 00b = Base Address register is 32 bits wide and can be mapped anywhere in the 32-bit Memory space All other encodings are ***reserved***.	RO	No	00b
3	**Prefetchable** 0 = Base Address register maps the USB 3382 Configuration registers into Non-Prefetchable Memory space ***Note:*** *Port 0 is hardwired to 0.*	RO	Yes	0
	Reserved	RsvdP	No	0
12:4	***Reserved***	RsvdP	No	0-0h
31:13	**Base Address 0** Base Address (**BAR0**) for the Device-Specific Memory-Mapped Configuration mechanism.	RW	Yes	0-0h
	Reserved	RsvdP	No	0-0h

Register 15-22. 14h Base Address 1 (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
0	**Memory Space Indicator** 0 = Base Address register maps the on-chip 8051/DMA Descriptor RAM into Memory space	RO	No	0
2:1	**Memory Map Type** 00b = Base Address register is 32 bits wide and can be mapped anywhere in the 32-bit Memory space 10b = Base Address register is 64 bits wide and can be mapped anywhere in the 64-bit Address space All other encodings are ***reserved***.	RO	No	00b
3	**Prefetchable** 0 = Base Address register maps the USB 3382 Configuration registers into Non-Prefetchable Memory space ***Note:*** *Port 0 is hardwired to 0.*	RO	Yes	0
15:4	***Reserved***	RsvdP	No	000h
31:16	**Base Address 1** Base Address (**BAR1**) for the Device-Specific Memory-Mapped Configuration mechanism. This Base Address field maps 32 KB of on-chip RAM used for storing 8051 microcode and/or on-chip DMA Descriptor tables. Accesses to this Memory space are limited to single DWord access for Writes, and up to 128 bytes for Reads.	RW	Yes	0000h

Register 15-23. 18h Base Address 2 (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
0	**Memory Space Indicator** 0 = Implemented as a Memory BAR	RO	No	0
2:1	**Memory Map Type** 00b = Base Address register is 32 bits wide and can be mapped anywhere in the 32-bit Memory space 10b = Base Address register is 64 bits wide and can be mapped anywhere in the 64-bit Address space All other encodings are ***reserved***.	RO	Yes	00b
3	**Prefetchable** 0 = Non-Prefetchable 1 = Prefetchable	RO	Yes	0
15:4	***Reserved***	RsvdP	No	000h
31:16	**Base Address 2** Resolution is 1 MB.	RW	Yes	0000h

Register 15-24. 1Ch Base Address 3 (USB Controller)

Bit(s)	Description		Type	Serial EEPROM and I²C	Default
Note: *This register has RW privilege if* ***BAR2/3*** *is configured as a 64-bit BAR (**Base Address 2** register Memory Map Type field (USB Controller, offset 18h[2:1]), is programmed to 10b).*					
0	**Memory Space Indicator** **BAR3** can be used as an independent 32-bit only BAR, or as the upper 32 bits of 64-bit **BAR2/3**. 0 = Memory space only supported	Offset 18h[2:1]=00b	RsvdP	No	0
		Offset 18h[2:1]=10b	RW	Yes	0
2:1	**Memory Map Type** 00b = Base Address register is 32 bits wide and can be mapped anywhere in the 32-bit Memory space All other encodings are ***reserved***.	Offset 18h[2:1]=00b	RsvdP	No	00b
		Offset 18h[2:1]=10b	RW	Yes	00b
3	**Prefetchable** 0 = Non-Prefetchable 1 = Prefetchable	Offset 18h[2:1]=00b	RsvdP	No	0
		Offset 18h[2:1]=10b	RW	Yes	0
15:4	***Reserved***	Offset 18h[2:1]=00b	RsvdP	No	000h
	When **BAR2/3** are used as a 64-bit BAR, bits [31:0] (including bits [15:4]) are used as the upper 32 bits.	Offset 18h[2:1]=10b	RW	Yes	000h
31:16	**Base Address 3**		RW	Yes	0000h

Register 15-25. 20h Base Address 4 (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
0	**Memory Space Indicator** 0 = Memory space only supported	RO	No	0
2:1	**Memory Map Type** 00b = Base Address register is 32 bits wide and can be mapped anywhere in the 32-bit Memory space 10b = Base Address register is 64 bits wide and can be mapped anywhere in the 64-bit Address space All other encodings are ***reserved***.	RO	Yes	00b
3	**Prefetchable** 0 = Non-Prefetchable 1 = Prefetchable	RO	Yes	0
15:4	***Reserved***	RsvdP	No	000h
31:16	**Base Address 4**	RW	Yes	0000h

Register 15-26. 24h Base Address 5 (USB Controller)

Bit(s)	Description		Type	Serial EEPROM and I^2C	Default
	Note: *This register has RW privilege if* ***BAR4/5*** *is configured as a 64-bit BAR (**Base Address 4** register Memory Map Type field (USB Controller, offset 20h[2:1]), is programmed to 10b).*				
0	**Memory Space Indicator** **BAR5** can be used as an independent 32-bit only BAR, or as the upper 32 bits of 64-bit **BAR4/5**. 0 = Memory space only supported	Offset 20h[2:1]=00b	RsvdP	No	0
		Offset 20h[2:1]=10b	RW	Yes	0
2:1	**Memory Map Type** 00b = Base Address register is 32 bits wide and can be mapped anywhere in the 32-bit Memory space All other encodings are ***reserved***.	Offset 20h[2:1]=00b	RsvdP	No	00b
		Offset 20h[2:1]=10b	RW	Yes	00b
3	**Prefetchable** 0 = Non-Prefetchable 1 = Prefetchable	Offset 20h[2:1]=00b	RsvdP	Yes	0
		Offset 20h[2:1]=10b	RW	Yes	0
15:4	***Reserved***	Offset 20h[2:1]=00b	RsvdP	No	000h
	When **BAR4/5** are used as a 64-bit BAR, bits [31:0] (including bits [15:4]) are used as the upper 32 bits.	Offset 20h[2:1]=10b	RW	Yes	000h
31:16	**Base Address 5**		RW	Yes	0000h

Register 15-27. 2Ch Subsystem ID and Subsystem Vendor ID (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
	Subsystem Vendor ID			
15:0	**Subsystem Vendor ID** Identifies the device manufacturer. Defaults to the PCI-SIG-issued Vendor ID of PLX (10B5h), if not overwritten by serial EEPROM and/or I^2C.	RO	Yes	10B5h
	Subsystem ID			
31:16	**Device ID** Identifies the particular device. Defaults to the PLX part number for the USB 3382, if not overwritten by serial EEPROM and/or I^2C.	RO	Yes	3382h

Register 15-28. 34h Capability Pointer (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
7:0	**Capability Pointer** Default 40h points to the **PCI Power Management Capability** structure.	RO	Yes	40h
31:8	***Reserved***	RsvdP	No	0000_00h

Register 15-29. 3Ch Bridge Control and PCI Interrupt Signal (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
PCI Interrupt Line				
7:0	**Interrupt Line Routing Value** The USB 3382 does ***not*** use this register; however, the register is included for operating system and device driver use.	RW	Yes	00h
PCI Interrupt Pin				
15:8	**Interrupt Pin** Identifies the Conventional PCI Interrupt Message(s) that the device (or device function) uses. Only value 00h or 01h is allowed in the USB 3382. 00h = Indicates that the device does not use Conventional PCI Interrupt Message(s) 01h, 02h, 03h, and 04h = Maps to Conventional PCI Interrupt Messages for INTA#, INTB#, INTC#, and INTD#, respectively	RO	Yes	01h
Min_Gnt				
23:16	**Minimum Grant** ***Reserved*** Does not apply to PCI Express.	RsvdP	No	00h
Max_Lat				
31:24	**Maximum Latency** ***Reserved*** Does not apply to PCI Express.	RsvdP	No	00h

15.8 PCI Power Management Capability Registers (Offsets 40h – 44h)

This section details the PCI Power Management Capability registers. Table 15-9 defines the register map.

Table 15-9. PCI Power Management Capability Register Map (All Ports and USB Controller)

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
PCI Power Management Capability		Next Capability Pointer (48h)	Capability ID (01h)	40h
PCI Power Management Data	PCI Power Management Control/Status Bridge Extensions ***(Reserved)***	PCI Power Management Status and Control		44h

Register 15-30. 40h PCI Power Management Capability (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
7:0	**Capability ID** Program to 01h, to indicate that the Capability structure is the **PCI Power Management Capability** structure.	RO	Yes	01h
15:8	**Next Capability Pointer** Default 48h points to the **MSI Capability** structure.	RO	Yes	48h
18:16	**Version** Default 011b indicates compliance with the *PCI Power Mgmt. r1.2*.	RO	Yes	011b
19	**PME Clock** Power Management Event (PME) clock. Does not apply to PCI Express. Returns a value of 0.	RsvdP	No	0
20	***Reserved***	RsvdP	No	0
21	**Device-Specific Initialization** 0 = Device-Specific Initialization is ***not*** required	RO	Yes	0
24:22	**AUX Current** The **Data** register (All Ports and USB Controller, offset 44h[31:24]) is not implemented, by default. Until serial EEPROM and/or I^2C writes a value, the **Data** register field is all zeros (0s). If serial EEPROM and/or I^2C writes to the **Data** register, the **Data** register indicates that it is implemented, and those agents can then Clear the AUX Current value. If the **Data** register is implemented: **1.** This field returns a value of 000b. **2.** The **Data** register takes precedence over this field. If wakeup from the D3cold state is not supported, this field returns a value of 000b.	RO	Yes	001b
25	**D1 Support** 1 = USB 3382 supports the D1 state	RO	No	1
26	**D2 Support** 1 = USB 3382 supports the D2 state	RO	No	1
31:27	**PME Support** Bits [31, 30, and 27] must be Set, to indicate that the USB 3382 will forward PME Messages, as required by the *PCI Express Base r2.1*.	RO	Yes	1Eh

Register 15-31. 44h PCI Power Management Status and Control (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
PCI Power Management Status and Control				
1:0	**Power State** Used to determine the current Device PM state of the Port/SuperSpeed USB, and to program the Port/SuperSpeed USB into a new Device PM state. 00b = D0 01b = D1 10b = D2 11b = D3hot If software attempts to write an unsupported state to this field, the Write operation completes normally; however, the data is discarded and no state change occurs.	RW	Yes	00b
2	***Reserved***	RsvdP	No	0
3	**No Soft Reset** 0 = D3hot to D0 state change causes a Fundamental Reset of the Port/SuperSpeed USB. This reset is propagated to downstream Ports and devices. 1 = Devices transitioning from the D3hot to D0 state, because of Power State commands, do not perform an internal reset.	RO	Yes	0
7:4	***Reserved***	RsvdP	No	0h

Register 15-31. 44h PCI Power Management Status and Control (All Ports and USB Controller) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
8	**PME Enable** 0 = Disables PME generation by the corresponding USB 3382 Port/ SuperSpeed USB 1 = Enables PME generation by the corresponding USB 3382 Port/ SuperSpeed USB	RWS	No	0
12:9	**Data Select** Initially writable by serial EEPROM and/or I^2C only[a]. This Configuration Space register (CSR) access privilege changes to RW after a Serial EEPROM and/or I^2C Write occurs to this register. Selects the field [14:13] (*Data Scale*) and **PCI Power Management Data** register *Data* field (All Ports and USB Controller, offset 44h[31:24]). 0h = D0 power consumed 1h = D1 power consumed 2h = D2 power consumed 3h = D3 power consumed All other encodings are ***reserved***.	RO	Yes	0h
14:13	**Data Scale** Writable by serial EEPROM and/or I^2C only[a]. Indicates the scaling factor to be used when interpreting the **PCI Power Management Data** register *Data* field (All Ports and USB Controller, offset 44h [31:24]) value. The value and meaning of the *Data Scale* field varies, depending upon which data value is selected by field [12:9] (*Data Select*). There are four internal *Data Scale* fields (one each, per *Data Select* values 0h, 1h, 2h, and 3h), per Port. For other *Data Select* values, the *Data* value returned is 00h.	RO	Yes	00b
15	**PME Status** 0 = PME is not generated by the corresponding USB 3382 Port/SuperSpeed USB 1 = PME is being generated by the corresponding USB 3382 Port/ SuperSpeed USB	RW1CS	No	0

Register 15-31. 44h PCI Power Management Status and Control (All Ports and USB Controller) ***(Cont.)***

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
PCI Power Management Control/Status Bridge Extensions				
21:16	***Reserved***	RsvdP	No	0-0h
22	**B2/B3 Support** ***Reserved*** Cleared, as required by the *PCI Power Mgmt. r1.2*.	RsvdP	No	0
23	**Bus Power/Clock Control Enable** ***Reserved*** Cleared, as required by the *PCI Power Mgmt. r1.2*.	RsvdP	No	0
PCI Power Management Data				
31:24	**Data** Writable by serial EEPROM and/or I^2C only[a]. There are four supported *Data Select* values (0h, 1h, 2h, and 3h), per Port/ SuperSpeed USB. For other *Data Select* values, the *Data* value returned is 00h. Selected by the **PCI Power Management Status and Control** register *Data Select* field (All Ports and USB Controller, offset 44h[12:9]).	RO	Yes	00h

a. *With no serial EEPROM nor previous I^2C programming, Reads return a value of 00h for the **PCI Power Management Status and Control** register Data Scale and **PCI Power Management Data** register Data fields (for all Data Selects).*

15.9 MSI Capability Registers (Offsets 48h – 64h)

This section details the Message Signaled Interrupt (MSI) Capability registers. Table 15-10 defines the register map.

Table 15-10. MSI Capability Register Map (All Ports and USB Controller)[a]

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
MSI Control	Next Capability Pointer (68h)	Capability ID (05h)	48h
MSI Address			4Ch
MSI Upper Address			50h
Reserved	MSI Data		54h
MSI Mask			58h
MSI Status			5Ch
Reserved			60h – 64h

a. *Offsets 54h, 58h, and 5Ch change to 50h, 54h, and 58h, respectively, when the **MSI Control** register MSI 64-Bit Address Capable bit (All Ports and USB Controller, offset 48h[23]) is Cleared.*

Register 15-32. 48h MSI Capability (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
MSI Capability Header				
7:0	**Capability ID** Program to 05h, as required by the *PCI r3.0.*	RO	Yes	05h
15:8	**Next Capability Pointer** Program to 68h, to point to the **PCI Express Capability** structure.	RO	Yes	68h
MSI Control				
16	**MSI Enable** 0 = MSIs for the Port/SuperSpeed USB are disabled 1 = MSIs for the Port/SuperSpeed USB are enabled, and INT*x* Interrupt Messages and PEX_INTA# output assertion are disabled	RW	Yes	0
19:17	**Multiple Message Capable** 000b = Port/SuperSpeed USB can request only one MSI Vector 001b = Port/SuperSpeed USB can request two MSI Vectors 010b through 111b = ***Reserved***	RO	Yes	001b
22:20	**Multiple Message Enable** 000b = Port/SuperSpeed USB is allocated one MSI Vector, by default 001b = Port/SuperSpeed USB is allocated two MSI Vectors 010b through 111b = ***Reserved*** ***Note:*** *This field should not be programmed with a value larger than that of field [19:17] (Multiple Message Capable). If the value of this field is larger than that of field [19:17], the Multiple Message Capable value takes effect.*	RW	Yes	000b
23	**MSI 64-Bit Address Capable** 0 = USB 3382 is capable of generating MSI 32-bit addresses (**MSI Address** register, offset 4Ch, is the Message address) 1 = USB 3382 is capable of generating MSI 64-bit addresses (**MSI Address** register, offset 4Ch, is the lower 32 bits of the Message address, and **MSI Upper Address** register, offset 50h, is the upper 32 bits of the Message address)	RO	Yes	1
24	**Per Vector Masking Capable** 0 = USB 3382 does not have Per Vector Masking capability 1 = USB 3382 has Per Vector Masking capability	RO	Yes	1
31:25	***Reserved***	RsvdP	No	0-0h

Register 15-33. 4Ch MSI Address (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
1:0	***Reserved***	RsvdP	No	00b
31:2	**Message Address** ***Note:*** *If the **MSI Control** register MSI 64-Bit Address Capable bit (All Ports and USB Controller, offset 48h[23]) is Set (default), refer to the **MSI Upper Address** register for the MSI Upper Address (All Ports and USB Controller, offset 50h).*	RW	Yes	0-0h

Register 15-34. 50h MSI Upper Address (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
31:0	**Message Upper Address** This register is valid/used only when the **MSI Control** register *MSI 64-Bit Address Capable* bit (All Ports and USB Controller, offset 48h[23]) is Set. MSI Write transaction upper address[63:32]. ***Note:*** *Refer to the **MSI Address** register for the MSI Lower Address (All Ports and USB Controller, offset 4Ch).*	RW	Yes	0000_0000h

Register 15-35. 54h MSI Data (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
Note: *The offset for this register changes from 54h, to 50h, when the **MSI Control** register MSI 64-Bit Address Capable bit (All Ports and USB Controller, offset 48h[23]) is Cleared.*				
15:0	**Message Data** MSI Write transaction TLP Payload.	RW	Yes	0000h
31:16	***Reserved***	RsvdP	No	0000h

Register 15-36. 58h MSI Mask (All Ports and USB Controller)

<table>
<tr><th>Bit(s)</th><th colspan="2">Description</th><th>Type</th><th>Serial EEPROM and I2C</th><th>Default</th></tr>
<tr><td colspan="6">Port/SuperSpeed USB interrupt sources are grouped into two categories – IRQSTAT1 and IRQSTAT0 registers (USB Controller, offsets 2Ch and 28h, respectively).
The quantity of MSI Vectors that are generated is determined by the MSI Control register Multiple Message Capable and Multiple Message Enable fields (All Ports and USB Controller, offset 48h[19:17 and 22:20], respectively):
• If one MSI Vector is enabled (default mode), both interrupt events are combined into a single Vector
• If two MSI Vectors are enabled, the individual vectors indicate:
– Vector[0] IRQSTAT1
– Vector[1] IRQSTAT0
Notes: The offset for this register changes from 58h, to 54h, when the MSI Control register MSI 64-Bit Address Capable bit (All Ports and USB Controller, offset 48h[23]) is Cleared.
The bits in this register can be used to mask their respective MSI Status register bits (All Ports and USB Controller, offset 5Ch).</td></tr>
<tr><td rowspan="2">0</td><td>MSI Mask for IRQSTAT1</td><td>Offset 48h[22:20]=001b</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td>MSI Mask for Shared Interrupt Sources
MSI mask for both interrupt sources when the MSI Control register Multiple Message Enable field indicates that the Host has allocated one Vector.</td><td>Offset 48h[22:20]=000b</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td rowspan="2">1</td><td>MSI Mask for IRQSTAT0</td><td>Offset 48h[22:20]=001b</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td>Reserved</td><td>Offset 48h[22:20]=000b</td><td>RsvdP</td><td>No</td><td>0</td></tr>
<tr><td>31:2</td><td colspan="2">Reserved</td><td>RsvdP</td><td>No</td><td>0-0h</td></tr>
</table>

Register 15-37. 5Ch MSI Status (All Ports and USB Controller)

<table>
<tr><th>Bit(s)</th><th colspan="2">Description</th><th>Type</th><th>Serial EEPROM and I2C</th><th>Default</th></tr>
<tr><td colspan="6">Port/SuperSpeed USB interrupt sources are grouped into two categories – IRQSTAT1 and IRQSTAT0 registers (USB Controller, offsets 2Ch and 28h, respectively).
The quantity of MSI Vectors that are generated is determined by the MSI Control register Multiple Message Capable and Multiple Message Enable fields (All Ports and USB Controller, offset 48h[19:17 and 22:20], respectively):
• If one MSI Vector is enabled (default mode), both interrupt events are combined into a single Vector
• If two MSI Vectors are enabled, the individual vectors indicate:
– Vector[0] IRQSTAT1
– Vector[1] IRQSTAT0
The bits in this register are self-Clearing, as soon as the USB 3382 generates the MSI Message. In the event that a particular MSI source is masked in the MSI Mask register (All Ports and USB Controller, offset 58h), Status bits will not be automatically Cleared if Message generation is masked or blocked.
Notes: The offset for this register changes from 5Ch, to 58h, when the MSI Control register MSI 64-Bit Address Capable bit (All Ports and USB Controller, offset 48h[23]) is Cleared.
The bits in this register can be masked by their respective MSI Mask register bits (All Ports and USB Controller, offset 58h).</td></tr>
<tr><td rowspan="2">0</td><td>MSI Pending Status for IRQSTAT1</td><td>Offset 48h[22:20]=001b</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td>MSI Pending Status for Shared Interrupt Sources
MSI pending status for both interrupt sources when the MSI Control register Multiple Message Enable field indicates that the Host has allocated one Vector.</td><td>Offset 48h[22:20]=000b</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td rowspan="2">1</td><td>MSI Pending Status for IRQSTAT0</td><td>Offset 48h[22:20]=001b</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td>Reserved</td><td>Offset 48h[22:20]=000b</td><td>RsvdP</td><td>No</td><td>0</td></tr>
<tr><td>31:2</td><td colspan="2">Reserved</td><td>RsvdP</td><td>No</td><td>0-0h</td></tr>
</table>

15.10 PCI Express Capability Registers (Offsets 68h – A0h)

This section details the PCI Express Capability registers. Command, Status, and Events are included in these registers. Table 15-11 defines the register map.

Note: *The Root Complex PCI Express Root Port registers, located at offsets 84h and 88h are visible/ available only in Root Complex mode. For further details, refer to Section 7.4.1.2, "PCI Express Root Port Registers."*

Table 15-11. PCI Express Capability Register Map

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
PCI Express Capability	Next Capability Pointer (00h)	Capability ID (10h)	68h
Device Capability			6Ch
Device Status	Device Control		70h
Link Capability (Legacy Adapter Mode) ***Reserved*** (Enhanced Adapter Mode)			74h
Link Status	Link Control		78h
Reserved			7Ch – 80h
Reserved (Adapter Mode) Root Capability and Control (Root Complex Mode)			84h
Reserved (Adapter Mode) Root Status (Root Complex Mode)			88h
Device Capability 2			8Ch
Device Status 2 ***(Reserved)***	Device Control 2		90h
Reserved			94h
Link Status 2 (Legacy Adapter Mode) ***Reserved*** (Enhanced Adapter Mode)	Link Control 2 (Legacy Adapter Mode) ***Reserved*** (Enhanced Adapter Mode)		98h
Reserved			9Ch – A0h

Register 15-38. 68h List and Capability (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
PCI Express Capability List				
7:0	**Capability ID** Program to 10h, as required by the *PCI Express Base r2.1*.	RO	Yes	10h
15:8	**Next Capability Pointer** 00h = This capability is the last capability in the USB 3382's Port/SuperSpeed USB's Capabilities list The USB 3382 Extended Capabilities list starts at offset 100h.	RO	Yes	00h
PCI Express Capability				
19:16	**Capability Version** The Ports/SuperSpeed USB program this field to 2h, as required by the *PCI Express Base r2.1*.	RO	Yes	2h
23:20	**Device/Port Type** Set at reset, as required by the *PCI Express Base r2.1*.	RO	Yes	1h
24	**Slot Implemented** ***Reserved***	RsvdP	No	0
29:25	**Interrupt Message Number** ***Reserved***	RsvdP	No	0-0h
31:30	***Reserved***	RsvdP	No	00b

Register 15-39. 6Ch Device Capability (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^{2}C	Default
2:0	**Maximum Payload Size Supported** 000b = Ports/SuperSpeed USB support a 128-byte maximum Payload 001b = Ports/SuperSpeed USB support a 256-byte maximum Payload No other encodings are supported.	RO	Yes	001b
4:3	**Phantom Functions Supported** ***Not supported***	RO	Yes	00b
5	**Extended Tag Field Supported** 0 = Maximum *Tag* field is 5 bits 1 = ***Reserved***	RO	Yes	0
8:6	**Endpoint L0s Acceptable Latency** 111b = Enables the capability	RO	Yes	111b
11:9	**Endpoint L1 Acceptable Latency** 111b = Enables the capability	RO	Yes	111b
14:12	***Reserved***, as required by the *PCI Express Base r2.1*.	RsvdP	No	000b
15	**Role-Based Error Reporting**	RO	Yes	1
17:16	***Reserved***	RsvdP	No	00b
25:18	**Captured Slot Power Limit Value** The upper limit on power supplied by the slot is determined by multiplying the value in this field by the value in field [27:26] (*Captured Slot Power Limit Scale*).	RO	Yes	00h
27:26	**Captured Slot Power Limit Scale** The upper limit on power supplied by the slot is determined by multiplying the value in this field by the value in field [25:18] (*Captured Slot Power Limit Value*). 00b = 1.0 01b = 0.1 10b = 0.01 11b = 0.001	RO	Yes	00b
31:28	***Reserved***	RsvdP	No	0h

Register 15-40. 70h Device Status and Control (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^{2}C	Default
	Device Control			
0	**Correctable Error Reporting Enable** 0 = Disables 1 = Enables the Port/SuperSpeed USB to report Correctable errors	RW	Yes	0
1	**Non-Fatal Error Reporting Enable** 0 = Disables 1 = Enables the Port/SuperSpeed USB to report Non-Fatal errors	RW	Yes	0
2	**Fatal Error Reporting Enable** 0 = Disables 1 = Enables the Port/SuperSpeed USB to report Fatal errors	RW	Yes	0
3	**Unsupported Request Reporting Enable** 0 = Disables 1 = Enables the Port/SuperSpeed USB to report UR errors	RW	Yes	0
4	**Enable Relaxed Ordering** **PCI Express Interface** ***Not supported***	RW	Yes	1
	USB Controller ***Not supported***	RsvdP	No	0
7:5	**Maximum Payload Size** Software can change this field to configure the Ports/SuperSpeed USB to support other Payload Sizes; however, software cannot change this field to a value larger than that indicated by the **Device Capability** register *Maximum Payload Size Supported* field (All Ports and USB Controller, offset 6Ch[2:0]). 000b = Port/SuperSpeed USB supports a 128-byte maximum Payload 001b = Port/SuperSpeed USB supports a 256-byte maximum Payload No other encodings are supported.	RW	Yes	000b
8	**Extended Tag Field Enable** ***Not supported***	RsvdP	No	0
9	**Phantom Functions Enable** ***Not supported***	RsvdP	No	0
10	**AUX Power PM Enable** 0 = Disables auxiliary power 1 = Enables auxiliary power	RWS	Yes	0
11	**Enable No Snoop** ***Not supported***	RsvdP	No	0

Register 15-40. 70h Device Status and Control (All Ports and USB Controller) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
14:12	**Ports (Type 0, Enhanced Adapter Mode)** ***Not supported*** Virtual PCI-to-PCI bridges do ***not*** generate Read Requests, they only forward them.	RsvdP	No	000b
	USB Controller (Type 1) **Max Read Request Size** Specifies the maximum PCI Express Read Request size, in bytes, that can be generated by the USB 3382, for DMA accesses. 000b = 128 bytes 001b = 256 bytes 010b = 512 bytes (default) 011b = 1,024 bytes 100b = 2,048 bytes 101b = 4,096 bytes All other encodings are ***reserved***.	RW	Yes	010b
15	***Reserved*** Hardwired to 0, as required by the *PCI Express Base r2.1*.	RsvdP	No	0
Device Status				
16	**Correctable Error Detected** Set when the Port/SuperSpeed USB detects a Correctable error, regardless of the bit 0 (*Correctable Error Reporting Enable*) state. 0 = Port/SuperSpeed USB did not detect a Correctable error 1 = Port/SuperSpeed USB detected a Correctable error	RW1C	Yes	0
17	**Non-Fatal Error Detected** Set when the Port/SuperSpeed USB detects a Non-Fatal error, regardless of the bit 1 (*Non-Fatal Error Reporting Enable*) state. 0 = Port/SuperSpeed USB did not detect a Non-Fatal error 1 = Port/SuperSpeed USB detected a Non-Fatal error	RW1C	Yes	0
18	**Fatal Error Detected** Set when the Port/SuperSpeed USB detects a Fatal error, regardless of the bit 2 (*Fatal Error Reporting Enable*) state. 0 = Port/SuperSpeed USB did not detect a Fatal error 1 = Port/SuperSpeed USB detected a Fatal error	RW1C	Yes	0
19	**Unsupported Request Detected** Set when the Port/SuperSpeed USB detects a UR, regardless of the bit 3 (*Unsupported Request Reporting Enable*) state. 0 = Port/SuperSpeed USB did not detect a UR 1 = Port/SuperSpeed USB detected a UR	RW1C	Yes	0
20	**AUX Power Detected** 1 = Auxiliary power is detected on the Port/SuperSpeed USB	ROS	Yes	1
21	**Transactions Pending**	RO	No	0
31:22	***Reserved***	RsvdP	No	0-0h

Register 15-41. 74h Link Capability (All Ports)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Note: *This register is visible/available only in Legacy Adapter mode.* ***Reserved*** *in Enhanced Adapter mode.*				
3:0	**Supported Link Speeds** Indicates the Port/SuperSpeed USB's supported Link speed. 0010b = 5.0 and 2.5 GT/s Link speeds are supported All other encodings are ***reserved***.	RO	Yes	0010b
9:4	**Maximum Link Width** The USB 3382 maximum Link width is x2 = 00_0010b. Valid widths are x1 or x2. Actual maximum Link width is defined by the STRAP_PORTCFG input. 00_0000b = ***Reserved*** 00_0001b = x1 00_0010b = x2 (Port 0 only) All other encodings are ***not supported***.	ROS	No	Defined by STRAP_PORTCFG input state, or programmed by serial EEPROM value for the **Port Configuration** register *Port Configuration* bit (Port 0, offset 574h[0])
11:10	**Active State Power Management (ASPM) Support** Active State Link PM support. Indicates the level of ASPM supported by the Port/SuperSpeed USB. 00b = ***Reserved*** 01b = L0s Link PM state entry is supported 10b = L1 ASPM is supported 11b = L0s and L1 Link PM states are supported	RO	Yes	11b

Register 15-41. 74h Link Capability (All Ports) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
14:12	**L0s Exit Latency** Indicates the L0s Link PM state exit latency for the given PCI Express Link. Value depends upon the **Advertised N_FTS** register *Advertised N_FTS* field (Port 0, offset B84h[7:0]) value and Link speed. Exit latency is calculated, as follows: • **2.5 GHz** – Multiply *Advertised N_FTS* x 4 (4 symbol times in 1 N_FTS) x 4 ns (1 symbol time at 2.5 GT/s) • **5.0 GHz** – Multiply *Advertised N_FTS* x 4 (4 symbol times in 1 N_FTS) x 2 ns (1 symbol time at 5.0 GT/s) 100b = Port/SuperSpeed USB's L0s Link PM state Exit Latency is 512 ns to less than 1 µs at 5.0 GT/s 101b = Port/SuperSpeed USB's L0s Link PM state Exit Latency is 1 µs to less than 2 µs at 2.5 GT/s All other encodings are ***reserved***.	RO	No	100b (5.0 GT/s) 101b (2.5 GT/s)
17:15	**L1 Exit Latency** Indicates the L1 Link PM state exit latency for the given PCI Express Link. Value depends upon the Link speed. 001b = Port/SuperSpeed USB's L1 Link PM state Exit Latency is 1 µs to less than 2 µs at 5.0 GT/s 010b = Port/SuperSpeed USB's L1 Link PM state Exit Latency is 2 µs to less than 4 µs at 2.5 GT/s All other encodings are ***reserved***.	RO	Yes	001b (5.0 GT/s) 010b (2.5 GT/s)
18	**Clock Power Management Capable** ***Not supported***	RsvdP	No	0
19	**Surprise Down Error Reporting Capable** ***Reserved***	RsvdP	No	0
20	**Data Link Layer Link Active Reporting Capable** ***Reserved***	RsvdP	No	0
21	**Link Bandwidth Notification Capability** ***Reserved***	RsvdP	No	0
23:22	***Reserved***	RsvdP	No	00b
31:24	**Port Number** ***Valid only in Enhanced Adapter mode. In Legacy Adapter mode, there are no Ports, only the Type 0 endpoint.*** The Port Number is defined by the STRAP_PORTCFG input. (Refer also to Table 15-12.) Available Port Numbers assigned by this field are 0 and 1. Port 2 is not affected by this register field.	ROS	No	Defined by STRAP_PORTCFG input state, or programmed by serial EEPROM value for the **Port Configuration** register *Port Configuration* bit (Port 0, offset 574h[0])

Table 15-12. Port/SuperSpeed Configurations

STRAP_PORTCFG Value	Port 0	Port 1	Port 2[a]	SuperSpeed USB
(default) 0	x1 Lane 0	x1 Lane 1	x1 Virtual Link	x1
1	x2 Lanes [0-1]		x1 Virtual Link	x1

a. Port 2 is an internal virtual PCI-to-PCI bridge that connects the USB Controller to the PCI Express fabric. It is not affected by the ***Link Capability*** *register Port Number field value.*

Register 15-42. 78h Link Status and Control (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Link Control				
1:0	**Active State Power Management (ASPM)** 00b = Disable[a] 01b = Enables only L0s Link PM state Entry 10b = Enables only L1 Link PM state Entry 11b = Enables both L0s and L1 Link PM state Entries	RW	Yes	00b
2	***Reserved***	RsvdP	No	0
3	**Read Request Return Parameter Control** Read Request Return Parameter "R" control. Read Completion Boundary (RCB). Cleared, as required by the *PCI Express Base r2.1*.	RW	Yes	0
4	**Link Disable** ***Reserved***	RsvdP	No	0
5	**Retrain Link** ***Reserved***	RsvdP	No	0
6	**Common Clock Configuration** 0 = Port/SuperSpeed USB and the device at the other end of the Port/SuperSpeed USB's PCI Express Link use an asynchronous Reference Clock source 1 = Port/SuperSpeed USB and the device at the other end of the Port/SuperSpeed USB's PCI Express Link use a common (synchronous) Reference Clock source (constant phase relationship)	RW	Yes	0
7	**Extended Sync** When Set, causes the Port/SuperSpeed USB to transmit: • 4,096 FTS Ordered-Sets in the L0s Link PM state, • Followed by a single SKIP Ordered-Set prior to entering the L0 Link PM state, • Finally, transmission of 1,024 TS1 Ordered-Sets in the *Recovery* state.	RW	Yes	0
8	**Clock Power Management Enable** ***Reserved*** Read and Writable only when the **Link Capability** register *Clock Power Management Capable* bit is Set.	RsvdP	No	0
9	**Hardware-Autonomous Width Disable** ***Reserved***	RsvdP	No	0
10	**Link Bandwidth Management Interrupt Enable** ***Reserved***	RsvdP	No	0
11	**Link Autonomous Bandwidth Interrupt Enable** ***Reserved***	RsvdP	No	0
15:12	***Reserved***	RsvdP	No	0h

Register 15-42. 78h Link Status and Control (All Ports and USB Controller) ***(Cont.)***

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Link Status				
19:16	**Current Link Speed** Indicates the current Link speed of the Port/SuperSpeed USB's PCI Express Link. 0001b = 2.5 GT/s Link speed 0010b = 5.0 GT/s Link speed All other encodings are ***reserved***. The value in this field is undefined when the Link is not up.	RO	No	0001b
25:20	**Negotiated Link Width** Dependent upon the physical Port configuration. Link width is determined by the negotiated value with the attached Lane/Port (Ports 0 and 1 only). 00_0001b = x1 or the Port/SuperSpeed USB is in the *DL_Down* state 00_0010b = x2 (Port 0 only) All other encodings are ***not supported***.	RO	No	00_0001b
26	***Reserved***	RsvdP	No	0
27	**Link Training** ***Reserved***	RsvdP	No	0
28	**Slot Clock Configuration** 0 = Indicates that the USB 3382 uses an independent clock 1 = Indicates that the USB 3382 uses the same physical Reference Clock that the platform provides on the connector	HwInit	Yes	0
29	**Data Link Layer Link Active** ***Reserved***	RsvdP	No	0
30	**Link Bandwidth Management Status** ***Reserved***	RsvdP	No	0
31	**Link Autonomous Bandwidth Status** ***Reserved***	RsvdP	No	0

a. *The Port Receiver must be capable of entering the L0s Link PM state, regardless of whether the state is disabled.*

Register 15-43. 84h Root Capability and Control (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
Note: *This register is visible/available only in Root Complex mode.*				
0	**System Error on Correctable Error Enable** 0 = Disables 1 = Enables the SuperSpeed USB to report Correctable System errors	RW	Yes	0
1	**System Error on Non-Fatal Error Reporting Enable** 0 = Disables 1 = Enables the SuperSpeed USB to report Non-Fatal System errors	RW	Yes	0
2	**System Error on Fatal Error Reporting Enable** 0 = Disables 1 = Enables the SuperSpeed USB to report Fatal System errors	RW	Yes	0
3	**PME Interrupt Enable**	RW	Yes	0
4	**CRS Software Invisibility Enable** ***Not supported***	RsvdP	No	0
15:5	***Reserved***	RsvdP	No	0-0h
16	**System Error On** ***Reserved***	RsvdP	No	0
31:17	***Reserved***	RsvdP	No	0-0h

Register 15-44. 88h Root Status (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
Note: *This register is visible/available only in Root Complex mode.*				
15:0	**PME Requester ID**	RO	Yes	0000h
16	**PME Status** 0 = PME is not generated by the Root Port 1 = PME is being generated by the Root Port	RW1C	Yes	0
17	**PME Pending**	RO	Yes	0
31:18	***Reserved***	RsvdP	No	0-0h

Register 15-45. 8Ch Device Capability 2 (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
3:0	**Completion Timeout Range Supported**	RO	Yes	0h
4	**Completion Timeout Range Supported**	RO	Yes	0
5	**ARI Forwarding Supported** ***Reserved***	RsvdP	No	0
31:6	***Reserved***	RsvdP	No	0-0h

Register 15-46. 90h Device Status and Control 2 (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
Device Control 2				
3:0	**Completion Timeout Value** Programmable timeout values include the values listed below. 00h = 20 ms 01h = 128 µs 02h = 2 ms 05h = 30 ms 06h = 200 ms 09h = 400 ms 10h = 2s 13h = 8s (default) 14h = 20s	RW	Yes	13h
4	**Completion Timeout Range Disable**	RW	Yes	0
5	**ARI Forwarding Enable** ***Reserved***	RsvdP	No	0
15:6	***Reserved***	RsvdP	No	0-0h
Device Status 2				
31:16	***Reserved***	RsvdP	No	0-0h

Register 15-47. 98h Link Status and Control 2 (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Note:	*This register is visible/available only in Legacy Adapter mode.* ***Reserved*** *in Enhanced Adapter mode.*			
	Link Control 2			
3:0	**Target Link Speed** 0001b = 2.5 GT/s Link speed is supported 0010b = 5.0 GT/s Link speed is supported All other encodings are ***reserved***.	RWS	Yes	0010b
4	**Enter Compliance**	RWS	Yes	0
5	**Hardware Autonomous Speed Disable** ***Reserved*** Initial transition to the highest supported common Link speed is not blocked by this bit.	RsvdP	No	0
6	**Selectable De-Emphasis** Selects the standard de-emphasis level when the Link is operating at 5.0 GT/s. When the Link is operating at 2.5 GT/s, the Setting of this bit has no effect (de-emphasis at 2.5 GT/s is -3.5 dB). 0 = -6 dB (Link is operating at 5.0 GT/s) 1 = -3.5 dB (Link is operating at 2.5 GT/s)	HwInit	Yes	0 (5.0 GT/s) 1 (2.5 GT/s)
9:7	**Transmit Margin** Intended for debug and compliance testing only.	RWS	Yes	000b
10	**Enter Modified Compliance** Intended for debug and compliance testing only. Valid only for Function 0 of Port 0.	RWS	Yes	0
11	**Compliance SOS** 1 = LTSSM must periodically send SKIP Ordered-Sets between sequences when sending the Compliance Pattern or Modified Compliance Pattern	RWS	Yes	0
12	**Compliance De-Emphasis** Sets the de-emphasis level in the *Polling.Compliance* substate, if the entry occurred due to bit 4 (*Enter Compliance*) being Set.	RWS	Yes	0
15:13	***Reserved***	RsvdP	No	000b
	Link Status 2			
16	**Current De-Emphasis Level** Reflects the de-emphasis level. 0 = -6 dB (Link is operating at 5.0 GT/s) 1 = -3.5 dB (Link is operating at 2.5 GT/s)	RO	Yes	0 (5.0 GT/s) 1 (2.5 GT/s)
31:17	***Reserved***	RsvdP	No	0-0h

15.11 Device Serial Number Extended Capability Registers (Offsets 100h – 134h)

This section details the Device Serial Number Extended Capability registers. Table 15-13 defines the register map.

Table 15-13. Device Serial Number Extended Capability Register Map (All Ports and USB Controller)

31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Next Capability Offset (FB4h)	Capability Version (1h)	PCI Express Extended Capability ID (0003h)	100h
Serial Number (Lower DW)			104h
Serial Number (Upper DW)			108h
Reserved			10Ch – 134h

Register 15-48. 100h Device Serial Number Extended Capability Header (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
15:0	**PCI Express Extended Capability ID** Program to 0003h, as required by the *PCI Express Base r2.1*.	RO	Yes	0003h
19:16	**Capability Version** Program to 1h, as required by the *PCI Express Base r2.1*.	RO	Yes	1h
31:20	**Next Capability Offset** Program to FB4h, which addresses the **Advanced Error Reporting Extended Capability** structure.	RO	Yes	FB4h

Register 15-49. 104h Serial Number (Lower DW) (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
31:0	**PCI Express Device Serial Number (1st DW)** Lower half of a 64-bit register. Value programmed by Serial EEPROM register initialization. Per the *PCI Express Base r2.1*, all USB 3382 Ports and the SuperSpeed USB must contain the same value; therefore, one physical register is shared by all Ports and the SuperSpeed USB. (Refer to Table 15-3.) The Serial Number registers contain the IEEE-defined 64-bit Extended Unique Identifier (EUI-64™), of which the upper 24 bits are the Company ID value (00_0EDFh) assigned by the IEEE Registration Authority, and the lower 40 bits are the Extension ID assigned by the identified Company (PLX). The most through least significant bytes are contained within the lowest through highest Byte addresses, respectively.	RO	Yes	B5DF_0E00h

Register 15-50. 108h Serial Number (Upper DW) (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
31:0	**PCI Express Device Serial Number (2nd DW)** Upper half of a 64-bit register. Value programmed by Serial EEPROM register initialization. Per the *PCI Express Base r2.1*, all USB 3382 Ports and the SuperSpeed USB must contain the same value; therefore, one physical register is shared by all Ports and the SuperSpeed USB. (Refer to Table 15-3.) The Serial Number registers contain the IEEE-defined 64-bit Extended Unique Identifier (EUI-64), of which the upper 24 bits are the Company ID value (00_0EDFh) assigned by the IEEE Registration Authority, and the lower 40 bits are the Extension ID assigned by the identified Company (PLX). The most through least significant bytes are contained within the lowest through highest Byte addresses, respectively.	RO	Yes	Silicon Revision AA: AA86_0210h Silicon Revision AB: AB86_0210h

15.12 Power Budget Extended Capability Registers (Offsets 138h – 144h)

This section details the Power Budget Extended Capability registers. These registers work differently than the others, especially with respect to serial EEPROM Reads and Writes. *For example*, when writing to Index 5 of the **Power Budget Data** register (All Ports and USB Controller, offset 140h), write 5 into the **Data Select** register *Data Select* field (All Ports and USB Controller, offset 13Ch[7:0]), then write the value into the **Power Budget Data** register. Table 15-14 defines the register map.

Table 15-14. Power Budget Extended Capability Register Map (All Ports and USB Controller)

31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Next Capability Offset (148h)	Capability Version (1h)	PCI Express Extended Capability ID (0004h)		138h
Reserved			Data Select	13Ch
Power Budget Data				140h
Power Budget Capability				144h

Register 15-51. 138h Power Budget Extended Capability Header (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
15:0	**PCI Express Extended Capability ID** Program to 0004h, as required by the *PCI Express Base r2.1*.	RO	Yes	0004h
19:16	**Capability Version** Program to 1h, as required by the *PCI Express Base r2.1*.	RO	Yes	1h
31:20	**Next Capability Offset** Program to 148h, which addresses the **Virtual Channel Extended Capability** structure.	RO	Yes	148h

Register 15-52. 13Ch Data Select (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
7:0	**Data Select** Indexes the Power Budget data reported, **Power Budget Data** registers, two per Port/SuperSpeed USB, and selects the DWord of Power Budget data that appears in each **Power Budget Data** register. Index values start at 0, to select the first DWord of Power Budget data; subsequent DWords of Power Budget data are selected by increasing index values 0 to 1.	RW	Yes	00h
31:8	***Reserved***	RsvdP	No	0000_00h

Register 15-53. 140h Power Budget Data (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
Note: *Two registers per Port/SuperSpeed USB can be programmed, through the serial EEPROM, I²C, and/or SMBus. Each non-zero register value describes the power usage for a different operating condition. Each configuration is selected by writing to the **Data Select** register Data Select field (All Ports and USB Controller, offset 13Ch[7:0]).*				
7:0	**Base Power** Two registers per Port/SuperSpeed USB. Specifies (in Watts) the base power value in the operating condition. This value must be multiplied by the field [9:8] (*Data Scale*) contents, to produce the actual power consumption value.	RO	Yes	00h
9:8	**Data Scale** Specifies the scale to apply to the Base Power value. The device power consumption is determined by multiplying the field [7:0] (*Base Power*) contents with the value corresponding to the encoding returned by this field. 00b = 1.0x 01b = 0.1x 10b = 0.01x 11b = 0.001x	RO	Yes	00b
12:10	**PM Sub-State** 000b = Power Management substate of the operating condition being described	RO	Yes	000b
14:13	**PM State** Power Management state of the operating condition being described. 00b = D0 state 01b = D1 state 10b = D2 state 11b = D3 state	RO	Yes	00b
17:15	**Type** Type of operating condition being described. 000b = PME Auxiliary 001b = Auxiliary 010b = Idle 011b = Sustained 111b = Maximum All other encodings are ***reserved***.	RO	Yes	000b
20:18	**Power Rail** Power Rail of the operating condition being described. 000b = Power 12V 001b = Power 3.3V 010b = Power 1.8V 111b = Thermal All other encodings are ***reserved***.	RO	Yes	000b
31:21	***Reserved***	RsvdP	No	0-0h

Register 15-54. 144h Power Budget Capability (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
0	**System Allocated** 1 = Power budget for the device is included within the system power budget	HwInit	Yes	1
31:1	***Reserved***	RsvdP	No	0-0h

15.13 Virtual Channel Extended Capability Registers (Offsets 148h – 1BCh)

This section details the Virtual Channel Extended Capability registers, which are duplicated for each Port. Table 15-15 defines the register map for one Port.

Table 15-15. Virtual Channel Extended Capability Register Map (All Ports and USB Controller)

31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Next Capability Offset (950h)	Capability Version (1h)	PCI Express Extended Capability ID (0002h)	148h
Port VC Capability 1			14Ch
Port VC Capability 2			150h
Port VC Status ***(Reserved)***		Port VC Control	154h
VC0 Resource Capability			158h
VC0 Resource Control			15Ch
VC0 Resource Status		***Reserved***	160h
Reserved			164h – 1BCh

Register 15-55. 148h Virtual Channel Extended Capability Header (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
15:0	**PCI Express Extended Capability ID** Program to 0002h, as required by the *PCI Express Base r2.1*.	RO	No	0002h
19:16	**Capability Version** Program to 1h, as required by the *PCI Express Base r2.1*.	RO	No	1h
31:20	**Next Capability Offset** Next extended capability is the **Vendor-Specific Extended Capability 2** structure, offset 950h.	RO	No	950h

Register 15-56. 14Ch Port VC Capability 1 (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
2:0	**Extended VC Counter** 000b = USB 3382 Port supports only one Virtual Channel, VC0 All other encodings are ***reserved***.	RO	Yes	000b
3	***Reserved***	RsvdP	No	0
6:4	**Low-Priority Extended VC Counter** ***Not supported***	RO	Yes	000b
7	***Reserved***	RsvdP	No	0
9:8	**Reference Clock** ***Reserved***	RsvdP	No	00b
11:10	**Port Arbitration Table Entry Size** ***Reserved***	RsvdP	Yes	10b
31:12	***Reserved***	RsvdP	No	0000_0h

Register 15-57. 150h Port VC Capability 2 (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
0	**VC Arbitration Capability** Indicates the type of VC arbitration supported by the device for the LPVC (Low-Priority Extended VC) group. 0 = Indicates that the Round-Robin (Hardware-Fixed) Arbitration scheme is not supported 1 = Indicates that the Round-Robin (Hardware-Fixed) Arbitration scheme is supported	RO	Yes	0
31:1	***Reserved***	RsvdP	No	0-0h

Register 15-58. 154h Port VC Status and Control (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
Port VC Control				
0	**Load VC Arbitration Table** ***Not supported*** Reads always return a value of 0.	RsvdP	No	0
3:1	**VC Arbitration Select** Selects the Port/SuperSpeed USB's VC arbitration type, as per the supported arbitration type indicated by the **Port VC Capability 2** register *VC Arbitration Capability* bit (All Ports and USB Controller, offset 150h[0]) value. 000b = Bit 0; Non-configurable (Hardware-Fixed) Arbitration All other encodings are ***reserved***.	RW	Yes	000b
15:4	***Reserved***	RsvdP	No	000h
Port VC Status				
16	**VC Arbitration Table Status** ***Reserved***	RsvdP	No	0
31:17	***Reserved***	RsvdP	No	0-0h

Register 15-59. 158h VC0 Resource Capability (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
7:0	**Port Arbitration Capability** Bit 0 = 1 – Non-configurable (Hardware-Fixed) Arbitration All other bits read 0.	RO	No	01h
14:8	***Reserved***	RsvdP	No	0-0h
15	**Reject Snoop Transactions** Not a PCI Express switch feature; therefore, this bit is Cleared.	RsvdP	No	0
22:16	**Maximum Time Slots** ***Reserved***	RsvdP	No	000_0000b
23	***Reserved***	RsvdP	No	0
31:24	**Port Arbitration Table Offset** Offset of the Port Arbitration Table, as the quantity of DQWords from the Base address of the **Virtual Channel Extended Capability** structure. 00h = Non-configurable (Hardware-Fixed) Arbitration All other encodings are ***reserved***.	RO	No	00h

Register 15-60. 15Ch VC0 Resource Control (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
0	**TC/VC Map** Defines Traffic Classes [7:0], respectively, and indicates which TCs are mapped to VC0. Traffic Class 0 (TC0) must be mapped to VC0. By default, Traffic Classes [7:1] are mapped to VC0.	RO	No	1
7:1		RW	Yes	7Fh
15:8	***Reserved***	RsvdP	No	00h
16	**Load Port Arbitration Table** Software writes this bit, to load the updated WRR Port Arbitration Table value to the internal logic. Software Read always returns a value of 0.	W1RZ	Yes	0
19:17	**Port Arbitration Select**	RW	Yes	000b
23:20	***Reserved***	RsvdP	No	0h
26:24	**VC0 ID** Defines the Port/SuperSpeed USB's VC0 ID code. 000b = VC0 (default; VC0 is the only/default VC) All other encodings are ***Reserved***.	RO	No	000b
30:27	***Reserved***	RsvdP	No	0h
31	**VC0 Enable** 0 = Not allowed 1 = Enables the Port/SuperSpeed USB's only/default VC, VC0	RO	No	1

Register 15-61. 160h VC0 Resource Status (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
15:0	***Reserved***	RsvdP	No	0000h
16	**Port/SuperSpeed USB Arbitration Table Status** The Port/SuperSpeed USB's Arbitration Table is ***not*** implemented; therefore, this bit has no function, and always reads 0.	RO	No	0
17	**VC0 Negotiation Pending** 0 = Port/SuperSpeed USB's VC0 negotiation is complete 1 = Port/SuperSpeed USB's VC0 initialization is not complete	RO	Yes	1
31:18	***Reserved***	RsvdP	No	0-0h

15.14 Device-Specific Registers (Offsets 1C0h – 444h)

This section details the Device-Specific registers located at offsets 1C0h through 444h. Device-Specific registers are unique to the USB 3382 and not referenced in the *PCI Express Base r2.1*. Table 15-16 defines the register map.

Other Device-Specific registers are detailed in Section 15.15, "Device-Specific Registers (Offsets 530h – B88h)."

Note: ***It is recommended that these registers not be changed from their default values***.

Table 15-16. Device-Specific Register Map (Offsets 1C0h – 444h)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Offset
Device-Specific Registers – Error Checking and Debug (Offsets 1C0h – 1FCh)	1C0h … 1FCh
Device-Specific Registers – Physical Layer (Offsets 200h – 25Ch)	200h … 25Ch
Device-Specific Registers – Serial EEPROM (Offsets 260h – 26Ch)	260h … 26Ch
Factory Test Only/Reserved	270h – 28Ch
Device-Specific Registers – I^{2}C and SMBus Slave Interfaces (Offsets 290h – 2C4h)	290h … 2C4h
Factory Test Only/Reserved	2C8h – 444h

15.14.1 Device-Specific Registers – Error Checking and Debug (Offsets 1C0h – 1FCh)

This section details the Device-Specific Error Checking and Debug registers. Table 15-17 defines the register map.

Table 15-17. Device-Specific Error Checking and Debug Register Map

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Offset
Factory Test Only	1C0h – 1C4h
ECC Error Check Disable	1C8h
Factory Test Only	1CCh – 1D4h
Clock Enable	1D8h
Debug Control	1DCh
Power Management Hot Plug User Configuration	1E0h
Egress Control and Status	1E4h
Bad TLP Counter	1E8h
Bad DLLP Counter	1ECh
Reserved	1F0h
Software Lane Status	1F4h
ACK Transmission Latency Limit	1F8h
Factory Test Only	1FCh

Register 15-62. 1C8h ECC Error Check Disable (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
3:0	***Factory Test Only***	RWS	Yes	0h
4	**Disable PCI Express INT*x* Message for Hot Plug Event-Triggered Interrupts** 0 = Hot Plug Event Interrupt Requests send an INT*x* Message to the PCI Express interface 1 = Hot Plug Event Interrupt Requests do not send an INT*x* Message to the PCI Express interface	RWS	Yes	0
5	***Factory Test Only***	RWS	Yes	0
6	**Enable PEX_INTA# Interrupt Output(s) for GPIO-Generated Interrupts** 0 = General-Purpose Input/Output (GPIO) Interrupt Requests send an INT*x* Message (and do not assert PEX_INTA#) 1 = GPIO Interrupt Requests assert PEX_INTA# (and do not send an INT*x* Message)	RWS	Yes	0
9:7	***Factory Test Only***	RWS	Yes	000b
10	***Factory Test Only***	RW1CS	Yes	0
31:11	***Reserved***	RsvdP	No	0-0h

Register 15-63. 1D8h Clock Enable (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Ports/SuperSpeed USB are automatically enabled, according to the Port configuration defined by the STRAP_PORTCFG input, which can be overridden by programming the **Port Configuration** register *Port Configuration* bit (Port 0, offset 574h[0]). An enabled Port can be selectively disabled, however, by Clearing the Port/SuperSpeed USB's *PDA* and *INT Port x/SuperSpeed USB Clock Enable* bits in this register. Port 0 must always remain enabled. ***Note:*** *It is not possible to enable more Ports/SuperSpeed USBs than the maximum specified for the device.*				
0	**PDA Port 0 Clock Enable** 0 = Disables 1 = Enables	RWS	Yes	1
1	**PDA Port 1 Clock Enable** 0 = Disables 1 = Enables	RWS	Yes	Defined by STRAP_PORTCFG input state, or programmed by serial EEPROM value for the **Port Configuration** register *Port Configuration* bit (Port 0, offset 574h[0])
2	**PDA Port 2 Clock Enable** 0 = Disables 1 = Enables	RWS	Yes	1
3	**PDA SuperSpeed USB Clock Enable** 0 = Disables 1 = Enables	RWS	Yes	1
4	**INT Port 0 Clock Enable** 0 = Disables 1 = Enables	RWS	Yes	1
5	**INT Port 1 Clock Enable** 0 = Disables 1 = Enables	RWS	Yes	Defined by STRAP_PORTCFG input state, or programmed by serial EEPROM value for the **Port Configuration** register *Port Configuration* bit (Port 0, offset 574h[0])
6	**INT Port 2 Clock Enable** 0 = Disables 1 = Enables	RWS	Yes	1
7	**INT SuperSpeed USB Clock Enable** 0 = Disables 1 = Enables	RWS	Yes	1

Register 15-63. 1D8h Clock Enable (Port 0) ***(Cont.)***

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
8	**REFCLK 1 Enable** Used to enable/disable Reference Clock Pair 1, PEX_REFCLK_OUTn1/ PEX_REFCLK_OUTp1. 0 = Disables 1 = Enables	RWS	Yes	1
9	**REFCLK 2 Enable** Used to enable/disable Reference Clock Pair 2, PEX_REFCLK_OUTn2/ PEX_REFCLK_OUTp2. 0 = Disables 1 = Enables	RWS	Yes	1
10	***Factory Test Only***	RWS	Yes	1
31:11	***Reserved***	RsvdP	No	0-0h

Register 15-64. 1DCh Debug Control (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Note: *This register* ***must*** *be the first Configuration register programmed by the serial EEPROM, at serial EEPROM locations 4h through 9h, as listed in Table 5-1, "Serial EEPROM Data Format."*				
0	**STRAP_DEBUG_SEL# Pin State** Reflects the logical state of the STRAP_DEBUG_SEL# input. 0 = Selected function is enabled	RO	No	0 (STRAP_DEBUG_SEL#=L) 1 (STRAP_DEBUG_SEL#=H)
1	***Reserved***	RsvdP	No	0
2	**STRAP_PROBE_MODE# Pin State** Reflects the logical state of the STRAP_PROBE_MODE# input. 0 = Selected function is enabled	RO	No	0 (STRAP_PROBE_MODE#=L) 1 (STRAP_PROBE_MODE#=H)
3	***Reserved***	RsvdP	No	0
4	**STRAP_UPCFG_TIMER_EN# Pin State** Reflects the logical state of the STRAP_UPCFG_TIMER_EN# input at reset. This bit's state can be subsequently overwritten by serial EEPROM and/or I^2C. 0 = Selected function is enabled	RO	Yes	0 (STRAP_UPCFG_TIMER_EN#=L) 1 (STRAP_UPCFG_TIMER_EN#=H)
5	**STRAP_SMBUS_EN# Pin State** Reflects the logical state of the STRAP_SMBUS_EN# input. 0 = SMBus Slave interface is enabled for device configuration (SMBus mode is enabled) 1 = SMBus Slave interface is disabled for device configuration (I^2C mode is enabled)	RO	Yes	0 (STRAP_SMBUS_EN#=L) 1 (STRAP_SMBUS_EN#=H)
6	**Force PCI Express Gen 1 (2.5 GT/s) Operation** 0 = PCI Express Links are forced to operate at Gen 1 bit rates 1 = PCI Express Links are allowed to transition to Gen 2 bit rates (5.0 GT/s) ***Note:*** *The value read from this bit is invalid.*	RWS	Yes	1
7	***Factory Test Only***	RWS	Yes	1
11:8	***Reserved***	RsvdP	No	0h
13:12	***Factory Test Only***	RWS	Yes	00b
14	***Factory Test Only***	RWS	Yes	0
15	***Reserved***	RsvdP	No	0

Register 15-64. 1DCh Debug Control (Port 0) ***(Cont.)***

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
16	**Upstream Hot Reset Control** 0 = Reset all logic, except Sticky bits and Device-Specific registers 1 = Reset only the Configuration Space registers of all Ports/SuperSpeed USB defined by the *PCI Express Base r2.1* ***Note:*** *Only a Fundamental Reset serial EEPROM load affects this bit.*	RWS	Yes	0
17	**Disable Serial EEPROM Load on Hot Reset** 0 = Enables serial EEPROM load upon Port 0 Hot Reset or *DL_Down* state 1 = Disables serial EEPROM load upon Port 0 Hot Reset or *DL_Down* state	RWS	Yes	0
19:18	***Reserved***	RsvdP	No	00b
20	**Upstream Port DL_Down Reset Propagation Disable** Setting this bit: • Enables Port 0 to ignore a Hot Reset training sequence, • Blocks the USB 3382 from manifesting an internal reset due to a DL_Down event, and • Prevents the downstream Ports from issuing a Hot Reset to downstream devices when a Hot Reset or DL_Down event occurs on the upstream Link	RWS	Yes	0
21	**Cut-Thru Enable** 0 = Disables Cut-Thru support 1 = Enables Cut-Thru support	RWS	Yes	1
22	**LANE_GOOD*x*#/GPIO*x* Pin Function Select** 0 = LANE_GOOD[1:0]# pins function as GPIO[1:0] pins. 1 = LANE_GOOD[1:0]# pins function as Lane Good status for Lanes [1-0], respectively. Additionally, GPIO3 functions as USB_LINK_GOOD#, which indicates whether the USB Link is up.	RWS	Yes	0
23	***Factory Test Only***	RWS	Yes	0
30:24	***Reserved***	RsvdP	No	0-0h
31	**8051 Cache Controller Off** Reads always return a value of 0. 0 = Enables 8051 instruction cache 1 = Disables 8051 instruction cache (default)	WO	Yes	1

Register 15-65. 1E0h Power Management Hot Plug User Configuration (All Ports and USB Controller)

Bit(s)	Description	Ports	Type	Serial EEPROM and I^2C	Default
Note: *When programming this register, always* • *Preserve the default states of bits [5:3]* • *Write 1 to bit 6*					
0	**L0s Entry Idle Counter** Traffic idle time to meet, to enter the L0s Link PM state. 0 = Idle condition must last 1 µs 1 = Idle condition must last 4 µs		RW	Yes	0
1	**ASPM L1 Disable**		RW	Yes	0
2	***Not enabled*** Functionality associated with this bit is enabled only on the downstream Ports.	Port 0	W1RZ	Yes	0
	HPC PME Turn-Off Enable 1 = PME Turn-Off Message is transmitted before the Port is turned Off on a downstream Port	Downstream	W1RZ	Yes	0
3	***Factory Test Only*** ***Note:*** *When programming this register, always preserve the state of this bit.*		W1RZ	Yes	0
4	***Factory Test Only*** ***Note:*** *When programming this register, always preserve the state of this bit.*		RWS	Yes	1
5	***Factory Test Only*** ***Note:*** *When programming this register, always preserve the state of this bit.*		RW	Yes	0
6	***Factory Test Only*** ***Note:*** *When programming this register, always write 1 to this bit.*		RW	Yes	1
7	**Disable PCI Express PM L1 Entry** 1 = Disables L1 Link PM state entry on Port 0, when Port 0 is placed into the D3hot state		RW	Yes	0
8	**DLLP Timeout Link Retrain Disable** Disable Link retraining when no Data Link Layer Packets (DLLPs) are received for more than 256 µs. 0 = Enables Link retraining when no DLLPs are received for more than 256 µs (default) 1 = DLLP Timeout is disabled		RW	Yes	0
9	***Factory Test Only***		RW	Yes	0
10	**L0s Entry Disable** 0 = Enables entry into the L0s Link PM state on a Port/the SuperSpeed USB when the L0s idle conditions are met 1 = Disables entry into the L0s Link PM state on a Port/the SuperSpeed USB when the L0s idle conditions are met		RW	Yes	0
11	***Factory Test Only***		RW	Yes	0

Register 15-65. 1E0h Power Management Hot Plug User Configuration (All Ports and USB Controller) *(Cont.)*

Bit(s)	Description	Ports	Type	Serial EEPROM and I^2C	Default
12	**PME on Host Resume Enable** 1 = Enables forwarding of PME Messages when USB Host Resume signaling is detected on the USB interface while the USB 3382 is in the D1, D2, or D3hot state		RW	Yes	0
15:13	***Factory Test Only***		RO	Yes	000b
17:16	**Present Pin Assignment Register** 01b = Port 1 All other encodings are ***reserved***.	Port 0	RO	Yes	01b
	Reserved	Downstream	RsvdP	No	00b
18	**HPC In-Band Presence Detect Disable** 1 = Disable Hot Plug Controller In-Band Presence Detect		RW	Yes	0
31:19	***Reserved***		RsvdP	No	0-0h

Register 15-66. 1E4h Egress Control and Status (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
1:0	***Reserved***	RW	Yes	00b
8:2	***Factory Test Only***	RW	Yes	20h
9	**Vendor-Specific Type 0 UR** 0 = Do not generate UR Vendor-Defined Type 0 Broadcast TLP in *DL_Down* state 1 = Generate UR Vendor-Defined Type 0 Broadcast TLP in *DL_Down* state	RW	Yes	0
10	**Egress Credit Timeout Enable** 0 = Egress Credit Timeout mechanism is disabled. 1 = Egress Credit Timeout mechanism is enabled. The timeout period is selected in bit 11 (*Egress Credit Timeout Value*). Status is reflected in bit 16 (*VC0 Egress Credit Timeout Status*). If the Egress Credit Timer is enabled and expires (due to lack of Flow Control credits from the connected device), the Port/SuperSpeed USB brings down its Link. This event generates a Surprise Down Uncorrectable error, for downstream Ports. For Port 0 Egress Credit Timeout, the connected upstream device detects the Surprise Down event.	RW	Yes	0
11	**Egress Credit Timeout Value** 0 = 384 to 512 ms 1 = 896 to 1,024 ms	RW	Yes	0
12	**Egress Credit Timeout Short** 0 = No Function 1 = Dependent upon the bit 11 (*Egress Credit Timeout Value*) value: • When bit 11 is Cleared, then timeout is 768 µs to 1 ms • When bit 11 is Set, then timeout is 1.792 ms to 2 ms	RW	Yes	0
15:13	***Reserved***	RsvdP	No	000b
16	**VC0 Egress Credit Timeout Status** 0 = No timeout 1 = Timeout	RW1C	No	0
18:17	**VC0 Egress Credit Timeout Type** 00b = Posted 01b = Non-Posted 10b = Completion 11b = ***Reserved***	RO	Yes	00b
31:19	***Reserved***	RsvdP	No	0-0h

Register 15-67. 1E8h Bad TLP Counter (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
31:0	**Bad TLP Counter** Counts the quantity of TLPs received with bad Link Cyclic Redundancy Check (LCRC), or quantity of TLPs with a Sequence Number Mismatch error. The Counter saturates at FFFF_FFFFh and does not roll over to 0000_0000h.	RWS	Yes	0000_0000h

Register 15-68. 1ECh Bad DLLP Counter (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
31:0	**Bad DLLP Counter** Counts the quantity of DLLPs received with bad LCRC, or quantity of DLLPs with a Sequence Number Mismatch error. The Counter saturates at FFFF_FFFFh and does not roll over to 0000_0000h.	RWS	Yes	0000_0000h

Register 15-69. 1F4h Software Lane Status (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
0	**Lane 0 Up Status** 0 = Lane is down 1 = Lane is up	RO	No	1
1	**Lane 1 Up Status** 0 = Lane is down 1 = Lane is up	RO	No	1
2	**Lane 2 Up Status** 0 = Lane is down 1 = Lane is up	RO	No	1
3	**SuperSpeed USB Up Status** 0 = SuperSpeed USB is down 1 = SuperSpeed USB is up	RO	No	1
31:4	***Reserved***	RsvdP	No	0000_000h

Register 15-70. 1F8h ACK Transmission Latency Limit (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
The value of this register should be valid after Link negotiation.				
11:0	**ACK Transmission Latency Limit** Acknowledge Control Packet (ACK) Transmission Latency Limit. The value of this field changes based upon Negotiated Link Width (All Ports and USB Controller, offset 78h[25:20]) encoding, after the Link is up. x1 Link width = FFh (STRAP_PORTCFG=L) x2 Link width = D9h (Port 0 only), (STRAP_PORTCFG=H)	RWS	Yes	Defined by STRAP_PORTCFG input state
15:12	***Reserved***	RsvdP	No	0h
23:16	**Upper 8 Bits of the Replay Timer Limit** If the serial EEPROM is not present, the value of this register changes based upon the negotiated Link width after the Link is up. The value in this register is a multiplier of the default internal timer values that are compliant to the *PCI Express Base r2.1*. These bits should normally remain the default value, 00h.	RWS	Yes	00h
30:24	***Reserved***	RsvdP	No	00h
31	**ACK Transmission Latency Timer Status** Indicates the written status of field [11:0] (*ACK Transmission Latency Limit*). After the register is written, either by software and/or serial EEPROM, this bit is Set, and Cleared only by a Fundamental Reset.	RO	No	0

15.14.2 Device-Specific Registers – Physical Layer (Offsets 200h – 25Ch)

This section details the Device-Specific Physical Layer (PHY) registers. Table 15-18 defines the register map.

Another Device-Specific PHY register is detailed in Section 15.15.6, "Device-Specific Registers – Physical Layer (Offsets B80h – B88h)."

Table 15-18. Device-Specific PHY Register Map (Offsets 200h – 25Ch) (Port 0)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Offset
Physical Layer Receiver Detect Status	Physical Layer Electrical Idle for Compliance Mask	200h
Physical Layer Receiver Not Detected Mask	Physical Layer Electrical Idle Detect Mask	204h
Factory Test Only		208h – 20Ch
Physical Layer User Test Pattern, Bytes 0 through 3		210h
Physical Layer User Test Pattern, Bytes 4 through 7		214h
Physical Layer User Test Pattern, Bytes 8 through 11		218h
Physical Layer User Test Pattern, Bytes 12 through 15		21Ch
Physical Layer Command and Status		220h
Physical Layer Function Control		224h
Physical Layer Test		228h
Factory Test Only		22Ch
Reserved	Physical Layer Port/SuperSpeed USB Command	230h
SKIP Ordered-Set Interval, Port and SuperSpeed USB Control		234h
SerDes Diagnostic Data		238h
Reserved		23Ch – 244h
Port Receiver Error Counters		248h
Reserved	Target Link Width (bits 7:0)	24Ch
Factory Test Only		250h
Physical Layer Additional Status		254h
PRBS Control/Status		258h
Factory Test Only		25Ch

Register 15-71. 200h Physical Layer Receiver Detect Status/Electrical Idle for Compliance Mask (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
This register is used for specifying whether the Lanes and SuperSpeed USB detected a Receiver during an LTSSM *Detect* state, but never detected an exit from Electrical Idle. When multiple bits are Set, and they correspond to Lanes that both belong to Port 0, either Lane can cause entry into the LTSSM *Polling.Compliance* substate.				
Physical Layer Electrical Idle for Compliance Mask				
This register allows masking that specifies whether the Lanes and SuperSpeed USB must never exit Electrical Idle, for entry to the LTSSM *Polling.Compliance* substate to occur.				
0	**Electrical Idle on SerDes 0 Causes Entry to Compliance State** When all the bits are Cleared, the LTSSM *Polling.Compliance* substate cannot be entered, due to the Electrical Idle condition. 1 = Lane 0 must have detected a Receiver during the LTSSM *Detect* state, and must not see an exit from Electrical Idle during the *Polling.Active* substate, to cause entry to the *Polling.Compliance* substate	RWS	Yes	1
1	**Electrical Idle on SerDes 1 Causes Entry to Compliance State** When all the bits are Cleared, the LTSSM *Polling.Compliance* substate cannot be entered, due to the Electrical Idle condition. 1 = Lane 1 must have detected a Receiver during the LTSSM *Detect* state, and must not see an exit from Electrical Idle during the *Polling.Active* substate, to cause entry to the *Polling.Compliance* substate	RWS	Yes	1
2	**Electrical Idle on SerDes 2 Causes Entry to Compliance State** When all the bits are Cleared, the LTSSM *Polling.Compliance* substate cannot be entered, due to the Electrical Idle condition. 1 = Lane 2 must have detected a Receiver during the LTSSM *Detect* state, and must not see an exit from Electrical Idle during the *Polling.Active* substate, to cause entry to the *Polling.Compliance* substate	RWS	Yes	1
3	**Electrical Idle on SuperSpeed USB Causes Entry to Compliance State** When all the bits are Cleared, the LTSSM *Polling.Compliance* substate cannot be entered, due to the Electrical Idle condition. 1 = The SuperSpeed USB must have detected a Receiver during the LTSSM *Detect* state, and must not see an exit from Electrical Idle during the *Polling.Active* substate, to cause entry to the *Polling.Compliance* substate	RWS	Yes	1
15:4	***Reserved***	RsvdP	No	000h

Register 15-71. 200h Physical Layer Receiver Detect Status/Electrical Idle for Compliance Mask (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Physical Layer Receiver Detect Status				
This register returns the Receiver's LTSSM *Detect* state status for both Lanes and SuperSpeed USB.				
16	**Receiver Detected on Lane 0** Returns the Receiver's LTSSM *Detect* state status, and reads back as 1 when a Receiver is detected on Lane 0.	RO	No	Set by SerDes
17	**Receiver Detected on Lane 1** Returns the Receiver's LTSSM *Detect* state status, and reads back as 1 when a Receiver is detected on Lane 1.	RO	No	Set by SerDes
18	**Receiver Detected on Lane 2** Returns the Receiver's LTSSM *Detect* state status, and reads back as 1 when a Receiver is detected on Lane 2.	RO	No	Set by SerDes
19	**Receiver Detected on SuperSpeed USB** Returns the Receiver's LTSSM *Detect* state status, and reads back as 1 when a Receiver is detected on the SuperSpeed USB Link.	RO	No	Set by SuperSpeed USB
31:20	***Reserved***	RsvdP	No	000h

Register 15-72. 204h Physical Layer Electrical Idle Detect/Receiver Detect Mask (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
This register is used to mask Electrical Idle Detect and Receiver functions for debug purposes. It can also be used to mask SerDes and SuperSpeed USB problems with these circuits. ***Note:*** *Masking Electrical Idle detect does not affect the inferred Electrical Idle detection.* ***Use the SerDes x and SuperSpeed USB Mask Electrical Idle Detect bits with care.***				
Physical Layer Electrical Idle Detect Mask				
Never Detect Electrical Idle mask. This register allows masking of the Electrical Idle Detect function, on a per-SerDes/SuperSpeed USB basis. When the bits in this register are Set, the Lane/SuperSpeed USB's *Electrical Idle* condition flag does not assert, regardless of the actual presence of Electrical Idle. Masking Electrical Idle detect does not affect the inferred Electrical Idle detection.				
0	**SerDes 0 Mask Electrical Idle Detect** 0 = Analog Electrical Idle detection is enabled for SerDes 0. 1 = Analog Electrical Idle detection is disabled for SerDes 0. Electrical Idle entrance inference, however, can be enabled by one or more available methods only if this bit is Set, and the **Physical Layer Function Control** register *Port 0 Electrical Idle Inference Disable* bit (Port 0, offset 224h[24]) is Cleared.	RWS	Yes	0
1	**SerDes 1 Mask Electrical Idle Detect** 0 = Analog Electrical Idle detection is enabled for SerDes 1. 1 = Analog Electrical Idle detection is disabled for SerDes 1. Electrical Idle entrance inference, however, can be enabled by one or more available methods only if this bit is Set, and the Port's **Physical Layer Function Control** register *Port x Electrical Idle Inference Disable* bit(s) (Port 0, offset 224h[25:24]) associated with SerDes 1 is Cleared.	RWS	Yes	0
2	**SerDes 2 Mask Electrical Idle Detect** 0 = Analog Electrical Idle detection is enabled for SerDes 2. 1 = Analog Electrical Idle detection is disabled for SerDes 2. Electrical Idle entrance inference, however, can be enabled by one or more available methods only if this bit is Set, and the **Physical Layer Function Control** register *Port 2 Electrical Idle Inference Disable* bit (Port 0, offset 224h[26]) is Cleared.	RWS	Yes	0
3	**SuperSpeed USB Mask Electrical Idle Detect** 0 = Analog Electrical Idle detection is enabled for the SuperSpeed USB. 1 = Analog Electrical Idle detection is disabled for the SuperSpeed USB. Electrical Idle entrance inference, however, can be enabled by one or more available methods only if this bit is Set, and the **Physical Layer Function Control** register *SuperSpeed USB Electrical Idle Inference Disable* bit (Port 0, offset 224h[27]) is Cleared.	RWS	Yes	0
15:4	***Reserved***	RsvdP	No	000h

Register 15-72. 204h Physical Layer Electrical Idle Detect/Receiver Detect Mask (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^{2}C	Default
Physical Layer Receiver Not Detected Mask				
Always Detect a Receiver mask. This register allows masking of the Receiver Detect function, on a per-SerDes/SuperSpeed USB basis. When the bits in this register are Set, the PHY functions as if the Lane/SuperSpeed USB detected a Receiver, regardless of the actual presence of a Receiver.				
16	**SerDes 0 Mask Receiver Not Detected** 1 = Masks the Receiver Not Detected for SerDes 0. Lane 0 will always detect a Receiver. The PHY functions as if a Receiver was detected on Lane 0, regardless of the actual presence of a Receiver.	RWS	Yes	0
17	**SerDes 1 Mask Receiver Not Detected** 1 = Masks the Receiver Not Detected for SerDes 1. Lane 1 will always detect a Receiver. The PHY functions as if a Receiver was detected on Lane 1, regardless of the actual presence of a Receiver.	RWS	Yes	0
18	**SerDes 2 Mask Receiver Not Detected** 1 = Masks the Receiver Not Detected for SerDes 2. Lane 2 will always detect a Receiver. The PHY functions as if a Receiver was detected on Lane 2, regardless of the actual presence of a Receiver.	RWS	Yes	0
19	**SuperSpeed USB Mask Receiver Not Detected** 1 = Masks the Receiver Not Detected for the SuperSpeed USB. The SuperSpeed USB will always detect a Receiver. The PHY functions as if a Receiver was detected on the SuperSpeed USB, regardless of the actual presence of a Receiver.	RWS	Yes	0
31:20	***Reserved***	RsvdP	No	000h

Register 15-73. 210h Physical Layer User Test Pattern, Bytes 0 through 3 (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Note: A 16-byte test pattern can be written to register offsets 210h through 21Ch. When User Test Pattern (UTP) transmission is enabled, Byte 0 of register offset 210h is transmitted first and Byte 3 (Byte 15 of the UTP) of register offset 21Ch is transmitted last. (Refer to Section 17.2.3, "Digital Loopback Master Mode," for further details.) Every byte of the UTP can be a Control or Data character. Illegal Control characters can be specified.				
7:0	Byte 0 of the UTP. This is the first byte transferred.	RW	Yes	00h
15:8	Byte 1 of the UTP.	RW	Yes	00h
23:16	Byte 2 of the UTP.	RW	Yes	00h
31:24	Byte 3 of the UTP.	RW	Yes	00h

Register 15-74. 214h Physical Layer User Test Pattern, Bytes 4 through 7 (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Note: A 16-byte test pattern can be written to register offsets 210h through 21Ch. When User Test Pattern (UTP) transmission is enabled, Byte 0 of register offset 210h is transmitted first and Byte 3 (Byte 15 of the UTP) of register offset 21Ch is transmitted last. (Refer to Section 17.2.3, "Digital Loopback Master Mode," for further details.) Every byte of the UTP can be a Control or Data character. Illegal Control characters can be specified.				
7:0	Byte 4 of the UTP. This is the fifth byte transferred.	RW	Yes	00h
15:8	Byte 5 of the UTP.	RW	Yes	00h
23:16	Byte 6 of the UTP.	RW	Yes	00h
31:24	Byte 7 of the UTP.	RW	Yes	00h

Register 15-75. 218h Physical Layer User Test Pattern, Bytes 8 through 11 (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
Note: *A 16-byte test pattern can be written to register offsets 210h through 21Ch. When User Test Pattern (UTP) transmission is enabled, Byte 0 of register offset 210h is transmitted first and Byte 3 (Byte 15 of the UTP) of register offset 21Ch is transmitted last. (Refer to Section 17.2.3, "Digital Loopback Master Mode," for further details.) Every byte of the UTP can be a Control or Data character. Illegal Control characters can be specified.*				
7:0	Byte 8 of the UTP. This is the ninth byte transferred.	RW	Yes	00h
15:8	Byte 9 of the UTP.	RW	Yes	00h
23:16	Byte 10 of the UTP.	RW	Yes	00h
31:24	Byte 11 of the UTP.	RW	Yes	00h

Register 15-76. 21Ch Physical Layer User Test Pattern, Bytes 12 through 15 (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
Note: *A 16-byte test pattern can be written to register offsets 210h through 21Ch. When User Test Pattern (UTP) transmission is enabled, Byte 0 of register offset 210h is transmitted first and Byte 3 (Byte 15 of the UTP) of register offset 21Ch is transmitted last. (Refer to Section 17.2.3, "Digital Loopback Master Mode," for further details.) Every byte of the UTP can be a Control or Data character. Illegal Control characters can be specified.*				
7:0	Byte 12 of the UTP. This is the thirteenth byte transferred.	RW	Yes	00h
15:8	Byte 13 of the UTP.	RW	Yes	00h
23:16	Byte 14 of the UTP.	RW	Yes	00h
31:24	Byte 15 of the UTP.	RW	Yes	00h

Register 15-77. 220h Physical Layer Command and Status (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
This register provides various Command and Status bits for Physical Layer operation.				
2:0	**Number of Ports Available** Returns the quantity of enabled Ports, based upon the selected Port configuration.	RO	No	Defined by STRAP_PORTCFG input state, or programmed by serial EEPROM value for the **Port Configuration** register *Port Configuration* bit (Port 0, offset 574h[0])
3	**Upstream Cross-Link Enable** 0 = Disables upstream cross-link, Port 0 cannot be connected to another upstream Port 1 = Enables upstream cross-link, Port 0 can be connected to another upstream Port	RWS	Yes	1
4	**Downstream Cross-Link Enable** 0 = Disables downstream cross-link, downstream Ports cannot be connected to other downstream Ports 1 = Enables downstream cross-link, downstream Ports can be connected to other downstream Ports	RWS	Yes	1
5	**Lane Reversal Disable** 1 = Disables Lane reversal on Port 0 only, when Port 0 is configured as a x2 Link width	RWS	Yes	0
6	***Reserved***	RsvdP	No	0
7	***Factory Test Only***	RWS	Yes	0
15:8	***Reserved***	RsvdP	No	00h
31:16	**User Test Pattern K-Code Flag** The corresponding UTP byte is transmitted as a kcode. ***Notice: Use caution when turning on k-characters, because the transmit logic does not examine illegal codes for validity. Also, sequences of control codes that can be detected as legal SKIP Ordered-Sets in the middle of the data pattern can confuse the Receive data checking logic. Therefore, it is recommended to not turn on k-characters when testing with a UTP.***	RWS	Yes	0000h

Register 15-78. 224h Physical Layer Function Control (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
This register contains Port/SuperSpeed USB-based PHY Safety bits.				
3:0	**Configuration Fail Counter[3:0]** Specifies the quantity of times that the *Configuration* state must fail, before the Port toggles its *Gen 2 Feature Disable* flag. Writing 0000b to this field disables this Gen 1 compatibility function. The initial value of this field is determined by the STRAP_UPCFG_TIMER_EN# input state. If the input is Low when reset de-asserts, the initial value of this field is 0001b; otherwise, the initial value is 0000b.	RWS	Yes	0000b (STRAP_UPCFG_TIMER_EN#=H) 0001b (STRAP_UPCFG_TIMER_EN#=L)
6:4	**Electrical Idle Inference Time Select[2:0]** Selects the amount of time to wait until no SKIP Ordered-Sets are detected, for Electrical Idle to be inferred. Does not affect Electrical Idle inference during the *Recovery.Speed* substate. 000b = 4 µs 001b = 6 µs 010b = 8 µs 011b = 16 µs 100b = 32 µs 101b = 64 µs 110b = 128 µs (default) 111b = 256 µs	RWS	Yes	110b
7	***Reserved***	RsvdP	No	0

Register 15-78. 224h Physical Layer Function Control (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
9:8	**Recovery.Speed Electrical Idle Inference Time Divider Select[1:0]** Selects the amount of time that no TS1 nor TS2 Ordered-Sets are detected during the *Recovery.Speed* substate, for Electrical Idle to be inferred. (Refer to the *PCI Express Base r2.1*, Section 4.2.4.3, for the specific Unit Interval (UI) values.) 00b = *PCI Express Base r2.1* UI 01b = *PCI Express Base r2.1* UI/2 10b = *PCI Express Base r2.1* UI/4 11b = *PCI Express Base r2.1* UI/8	RWS	Yes	00b
11:10	**Detect.Quiet Wait Time Select Code[1:0]** Selects the amount of time to wait during the *Detect.Quiet* substate, before starting the Receiver Detect operation, when a break from Electrical Idle is detected. If Electrical Idle is detected on the Lanes and SuperSpeed USB, the wait time is 12 ms. 00b = 0 ms 01b = 4 ms 10b = 8 ms 11b = 12 ms	RWS	Yes	00b
15, 13:12	**Unconditional SerDes/SuperSpeed USB Disable**	RWS	Yes	000b
14	***Reserved***	RsvdP	No	0
16	**Inferred Electrical Idle Inference Exit Type** Selects the method used to detect exit from Electrical Idle, after Electrical Idle has been inferred. 0 = **Fast Method** – Type 0 Exit mode is used, which uses conventional analog Electrical Idle Exit Detection circuitry 1 = **Slow Method** – Type 1 Exit mode is used, which uses the Symbol Framer Detection Time Select Code and Inferred Electrical Idle Exit Time Select Code Timers (fields [21:20 and 19:18], respectively)	RWS	Yes	0
17	***Reserved***	RsvdP	No	0

Register 15-78. 224h Physical Layer Function Control (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
19:18	**Inferred Electrical Idle Exit Time Select Code** When Electrical Idle has been inferred and the Electrical Idle Inference Exit Type is 1 (bit 16 is Set), this field selects the amount of time that the SerDes/SuperSpeed USB Receive Data path remains disabled. 00b = 2 µs 01b = 4 µs 10b = 8 µs 11b = 16 µs	RWS	Yes	00b
21:20	**Symbol Framer Detection Time Select Code** When Electrical Idle has been inferred and the Electrical Idle Inference Exit Type is 1 (bit 16 is Set), this field selects the amount of time that the symbol framer is allowed to obtain symbol lock. 00b = 128 ns 01b = 256 ns 10b = 512 ns 11b = 1 µs	RWS	Yes	00b
23:22	***Reserved***	RsvdP	No	00b

Register 15-78. 224h Physical Layer Function Control (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
24	**Port 0 Electrical Idle Inference Disable** 0 = Electrical Idle inference is enabled, if the **Physical Layer Electrical Idle Detect Mask** register *SerDes x Mask Electrical Idle Detect* bit(s) (Port 0, offset 204h[1:0]) are Set, for the SerDes associated with Port 0. 1 = Overall Electrical Idle inference logic is disabled on Port 0. Electrical Idle inference during the *Recovery.Speed* substate is not affected and will continue to operate.	RWS	Yes	0
25	**Port 1 Electrical Idle Inference Disable** 0 = Electrical Idle inference is enabled, if the **Physical Layer Electrical Idle Detect Mask** register *SerDes 1 Mask Electrical Idle Detect* bit (Port 0, offset 204h[1]) is Set. 1 = Overall Electrical Idle inference logic is disabled on Port 1. Electrical Idle inference during the *Recovery.Speed* substate is not affected and will continue to operate.	RWS	Yes	0
26	**Port 2 Electrical Idle Inference Disable** 0 = Electrical Idle inference is enabled, if the **Physical Layer Electrical Idle Detect Mask** register *SerDes 2 Mask Electrical Idle Detect* bit (Port 0, offset 204h[2]) is Set. 1 = Overall Electrical Idle inference logic is disabled on Port 2. Electrical Idle inference during the *Recovery.Speed* substate is not affected and will continue to operate.	RWS	Yes	0
27	**SuperSpeed USB Electrical Idle Inference Disable** 0 = Electrical Idle inference is enabled, if the **Physical Layer Electrical Idle Detect Mask** register *SuperSpeed USB Mask Electrical Idle Detect* bit (Port 0, offset 204h[3]) is Set. 1 = Overall Electrical Idle inference logic is disabled on the SuperSpeed USB. Electrical Idle inference during the *Recovery.Speed* substate is not affected and will continue to operate.	RWS	Yes	0

Register 15-78. 224h Physical Layer Function Control (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
28	**Port 0 Electrical Idle Inference on EIOS Receipt Enable** 0 = Electrical Idle inference is enabled upon Electrical Idle Ordered-Set (EIOS) receipt, if the **Physical Layer Electrical Idle Detect Mask** register *SerDes x Mask Electrical Idle Detect* bit(s) (Port 0, offset 204h[1:0]) are Set, for the SerDes associated with Port 0 1 = Electrical Idle will be inferred as soon as an EIOS is received on any Lane associated with Port 0	RWS	Yes	0
29	**Port 1 Electrical Idle Inference on EIOS Receipt Enable** 0 = Electrical Idle inference is enabled upon EIOS receipt, if the **Physical Layer Electrical Idle Detect Mask** register *SerDes 1 Mask Electrical Idle Detect* bit (Port 0, offset 204h[1]) is Set. 1 = Electrical Idle will be inferred as soon as an EIOS is received on Lane 1	RWS	Yes	0
30	**Port 2 Electrical Idle Inference on EIOS Receipt Enable** 0 = Electrical Idle inference is enabled upon EIOS receipt, if the **Physical Layer Electrical Idle Detect Mask** register *SerDes 2 Mask Electrical Idle Detect* bit (Port 0, offset 204h[2]) is Set 1 = Electrical Idle will be inferred as soon as an EIOS is received on Lane 2	RWS	Yes	0
31	**SuperSpeed USB Electrical Idle Inference on EIOS Receipt Enable** 0 = Electrical Idle inference is enabled upon EIOS receipt, if the **Physical Layer Electrical Idle Detect Mask** register *SuperSpeed USB Mask Electrical Idle Detect* bit (Port 0, offset 204h[3]) is Set 1 = Electrical Idle will be inferred as soon as an EIOS is received on the SuperSpeed USB	RWS	Yes	0

Register 15-79. 228h Physical Layer Test (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
This register provides controls to enable various PHY test modes.				
0	**Port 0 Timer Test Mode Enable** 0 = Normal PHY Timer parameters are used 1 = Millisecond scale timers in the Port 0 LTSSM are reduced to microsecond scale	RWS	Yes	0
1	**Port 1 Timer Test Mode Enable** 0 = Normal PHY Timer parameters are used 1 = Millisecond scale timers in the Port 1 LTSSM are reduced to microsecond scale	RWS	Yes	0
2	**Port 2 Timer Test Mode Enable** 0 = Normal PHY Timer parameters are used 1 = Millisecond scale timers in the Port 2 LTSSM are reduced to microsecond scale	RWS	Yes	0
3	**SuperSpeed USB Timer Test Mode Enable** 0 = Normal PHY Timer parameters are used 1 = Millisecond scale timers in the SuperSpeed USB LTSSM are reduced to microsecond scale	RWS	Yes	0
4	**SKIP Timer Test Mode Enable**	RW	Yes	0
5	**Ignore Compliance Receive TCB**	RWS	Yes	0
6	**Analog Loopback Enable** 0 = USB 3382 enters Digital Loopback Slave mode if an external device sends at least two consecutive TS1 Ordered-Sets that have the *Loopback* bit exclusively Set in the TS1 Training Control symbol. The USB 3382 then loops back data through the Elastic buffer, 8b/10b decoder, and 8b/10b encoder. 1 = Loopback point of all Ports/SuperSpeed USB is located before the Elastic buffer. This means that data recovered from the Serial data in the recovered Receive Clock domain is re-serialized, then re-transmitted in that same recovered clock domain. This allows the Loopback Master to transmit and receive a user test pattern (UTP) in an asynchronous clocking system. It is also the **required** mode for re-transmitting a PRBS pattern back to the Loopback Master. Overrides the Lane/SuperSpeed USB's Parallel "Digital" Loopback Setting bit(s) [27:24] (*Lane x/SuperSpeed USB Parallel Loopback Path Enable*).	RWS	Yes	0
7	***Reserved***	RsvdP	No	0

Register 15-79. 228h Physical Layer Test (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
8	***Factory Test Only***	RW	Yes	0
9	***Factory Test Only***	RO	No	0
10	***Factory Test Only***	RW1C	Yes	0
19:11	***Reserved***	RsvdP	No	0-0h
20	**Lane 0 Serial Loopback Path Enable** 0 = Disabled 1 = Lane 0 enables the Serial *Loopback* (Master) path, regardless of the LTSSM state	RW	Yes	0
21	**Lane 1 Serial Loopback Path Enable** 0 = Disabled 1 = Lane 1 enables the Serial *Loopback* (Master) path, regardless of the LTSSM state	RW	Yes	0
22	**Lane 2 Serial Loopback Path Enable** 0 = Disabled 1 = Lane 2 enables the Serial *Loopback* (Master) path, regardless of the LTSSM state	RW	Yes	0
23	**SuperSpeed USB Serial Loopback Path Enable** 0 = Disabled 1 = SuperSpeed USB enables the Serial *Loopback* (Master) path, regardless of the LTSSM state	RW	Yes	0

Register 15-79. 228h Physical Layer Test (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
24	**Lane 0 Parallel Loopback Path Enable** It is recommended that Port 0 be placed into a *Port Disable* state, by Setting the **Port and SuperSpeed USB Control** register *Disable Port 0* bit (Port 0, offset 234h[16]), followed by a *Port Quiet* state, by Setting the *Port 0 Quiet* bit (Port 0, offset 234h[20]), before Setting this bit. 0 = Parallel "Digital" *Loopback* (Slave) path is disabled for this Lane 1 = Lane manually enables the Parallel "Digital" *Loopback* (Slave) path, regardless of the LTSSM state ***Note:*** *This path is automatically enabled when the LTSSM enters the Loopback.Active substate, as a Loopback Slave.*	RW	Yes	0
25	**Lane 1 Parallel Loopback Path Enable** It is recommended that the Port associated with this Lane (Port 0 or Port 1, as appropriate) be placed into a *Port Disable* state, by Setting the Port's **Port and SuperSpeed USB Control** register *Disable Port x* bit(s) (Port 0, offset 234h[17:16]), followed by a *Port Quiet* state, by Setting the Port's *Port x Quiet* bit (Port 0, offset 234h[21:20]), before Setting this bit. 0 = Parallel "Digital" *Loopback* (Slave) path is disabled for this Lane 1 = Lane manually enables the Parallel "Digital" *Loopback* (Slave) path, regardless of the LTSSM state ***Note:*** *This path is automatically enabled when the LTSSM enters the Loopback.Active substate, as a Loopback Slave.*	RW	Yes	0
26	**Lane 2 Parallel Loopback Path Enable** It is recommended that Port 2 be placed into a *Port Disable* state, by Setting the **Port and SuperSpeed USB Control** register *Disable Port 2* bit (Port 0, offset 234h[18]), followed by a *Port Quiet* state, by Setting the *Port 2 Quiet* bit (Port 0, offset 234h[22]), before Setting this bit. 0 = Parallel "Digital" *Loopback* (Slave) path is disabled for this Lane 1 = Lane manually enables the Parallel "Digital" *Loopback* (Slave) path, regardless of the LTSSM state ***Note:*** *This path is automatically enabled when the LTSSM enters the Loopback.Active substate, as a Loopback Slave.*	RW	Yes	0
27	**SuperSpeed USB Parallel Loopback Path Enable** It is recommended that the SuperSpeed USB be placed into a *Disable* state, by Setting the **Port and SuperSpeed USB Control** register *Disable SuperSpeed USB* bit (Port 0, offset 234h[19]), followed by a *Quiet* state, by Setting the *SuperSpeed USB Quiet* bit (Port 0, offset 234h[23]), before Setting this bit. 0 = Parallel "Digital" *Loopback* (Slave) path is disabled for the SuperSpeed USB 1 = SuperSpeed USB manually enables the Parallel "Digital" *Loopback* (Slave) path, regardless of the LTSSM state ***Note:*** *This path is automatically enabled when the LTSSM enters the Loopback.Active substate, as a Loopback Slave.*	RW	Yes	0

Register 15-79. 228h Physical Layer Test (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
28	**SerDes 0 User Test Pattern Enable** 0 = Disables user test pattern transmission 1 = Enables user test pattern transmission ***Notes:*** ***This bit and the SerDes Test register SerDes 0 BIST Generator/ Checker Enable bit (Port 0, offset B88h[16]) are mutually exclusive functions and must not be enabled together for SerDes 0. The logical result of both bits ANDed with one another must be 0.*** ***UTP transmission should be enabled only when operating as a Loopback Master, or when the LTSSM has returned to the Detect.Quiet substate and the Port and SuperSpeed USB Control register Port 0 Quiet bit (Port 0, offset 234h[20]) is Set.***	RW	Yes	0
29	**SerDes 1 User Test Pattern Enable** 0 = Disables user test pattern transmission 1 = Enables user test pattern transmission ***Notes:*** ***This bit and the SerDes Test register SerDes 1 BIST Generator/ Checker Enable bit (Port 0, offset B88h[17]) are mutually exclusive functions and must not be enabled together for SerDes 1. The logical result of both bits ANDed with one another must be 0.*** ***UTP transmission should be enabled only when operating as a Loopback Master, or when the LTSSM has returned to the Detect.Quiet substate and the Port's Port and SuperSpeed USB Control register Port x Quiet bit (Port 0, offset 234h[21:20]) is Set.***	RW	Yes	0
30	**SerDes 2 User Test Pattern Enable** 0 = Disables user test pattern transmission 1 = Enables user test pattern transmission ***Notes:*** ***This bit and the SerDes Test register SerDes 2 BIST Generator/ Checker Enable bit (Port 0, offset B88h[18]) are mutually exclusive functions and must not be enabled together for SerDes 2. The logical result of both bits ANDed with one another must be 0.*** ***UTP transmission should be enabled only when operating as a Loopback Master, or when the LTSSM has returned to the Detect.Quiet substate and the Port and SuperSpeed USB Control register Port 2 Quiet bit (Port 0, offset 234h[22]) is Set.***	RW	Yes	0
31	**SuperSpeed USB User Test Pattern Enable** 0 = Disables user test pattern transmission 1 = Enables user test pattern transmission ***Notes:*** ***This bit and the SerDes Test register SuperSpeed USB BIST Generator/Checker Enable bit (Port 0, offset B88h[19]) are mutually exclusive functions and must not be enabled together for the SuperSpeed USB. The logical result of both bits ANDed with one another must be 0.*** ***UTP transmission should be enabled only when operating as a Loopback Master, or when the LTSSM has returned to the Detect.Quiet substate and the Port and SuperSpeed USB Control register SuperSpeed USB Quiet bit (Port 0, offset 234h[23]) is Set.***	RW	Yes	0

Register 15-80. 230h Physical Layer Port/SuperSpeed USB Command (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
This register provides the Port/SuperSpeed USB Loopback, Scrambler Disable, and Compliance Receive commands, and Ready as Loopback Master status.				
0	**Port 0 Loopback Command** 0 = Port 0 is not enabled to go to the *Loopback* Master state. 1 = Port 0 attempts to enter the *Loopback* state as a Loopback Master. If this bit is Set before the *Configuration* state is reached, the *Configuration.Linkwidth.Start* to *Loopback* path is used. If this bit is Set later, the *Recovery.Idle* to *Loopback* path is used.	RWS	Yes	0
1	**Port 0 Scrambler Disable Command** When Set, unconditionally disables the data scramblers on either Lane associated with Port 0, and causes the *Disable Scrambling* Training Control Bit to be Set in the transmitted Training Sets. If a serial EEPROM load Sets this bit, the scrambler is disabled in a *Configuration.Complete* substate. If software Sets this bit when the Link is in the Up state, hardware disables its scrambler without executing the Link Training protocol. This scrambler disable takes effect after the Link passes through *Configuration* again. The upstream/downstream device scrambler will not be disabled. 0 = Port 0's scrambler is enabled 1 = Port 0's scrambler is disabled	RWS	Yes	0
2	**Port 0 Compliance Receive Command** 0 = When Port 0 transmits TS1 Ordered-Sets, the *Compliance Receive* Training Control Bit within these Ordered-Sets is not Set during the *Polling.Active* nor *Loopback.Entry* substate 1 = When Port 0 transmits TS1 Ordered-Sets, the *Compliance Receive* Training Control Bit within these Ordered-Sets is Set during the *Polling.Active* or *Loopback.Entry* substate	RWS	Yes	0
3	**Port 0 Ready as Loopback Master** Link Training and Status State Machine (LTSSM) established Loopback as a Master for Port 0. 0 = Port 0 is not in Loopback Master mode. 1 = Indicates that Port 0 has successfully transitioned to the *Loopback.Active* substate as a Loopback Master. The LTSSM remains in this substate, until bit 0 (*Port 0 Loopback Command*) is Cleared. This bit is Cleared when the USB 3382 exits the *Loopback.Active* substate.	RO	No	0

Register 15-80. 230h Physical Layer Port/SuperSpeed USB Command (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^{2}C	Default
4	**Port 1 Loopback Command** 0 = Port 1 is not enabled to go to the *Loopback* Master state. 1 = Port 1 attempts to enter the *Loopback* state as a Loopback Master. If this bit is Set before the *Configuration* state is reached, the *Configuration.Linkwidth.Start* to *Loopback* path is used. If this bit is Set later, the *Recovery.Idle* to *Loopback* path is used.	RWS	Yes	0
5	**Port 1 Scrambler Disable Command** When Set, unconditionally disables the data scramblers on Lane 1, and causes the *Disable Scrambling* Training Control Bit to be Set in the transmitted Training Sets. If a serial EEPROM load Sets this bit, the scrambler is disabled in a *Configuration.Complete* substate. If software Sets this bit when the Link is in the Up state, hardware disables its scrambler without executing the Link Training protocol. This scrambler disable takes effect after the Link passes through *Configuration* again. The upstream/downstream device scrambler will not be disabled. 0 = Port 1's scrambler is enabled 1 = Port 1's scrambler is disabled	RWS	Yes	0
6	**Port 1 Compliance Receive Command** 0 = When Port 1 transmits TS1 Ordered-Sets, the *Compliance Receive* Training Control Bit within these Ordered-Sets is not Set during the *Polling.Active* nor *Loopback.Entry* substate 1 = When Port 1 transmits TS1 Ordered-Sets, the *Compliance Receive* Training Control Bit within these Ordered-Sets is Set during the *Polling.Active* or *Loopback.Entry* substate	RWS	Yes	0
7	**Port 1 Ready as Loopback Master** LTSSM established Loopback as a Master for Port 1. 0 = Port 1 is not in Loopback Master mode. 1 = Indicates that Port 1 has successfully transitioned to the *Loopback.Active* substate as a Loopback Master. The LTSSM remains in this substate, until bit 4 (*Port 1 Loopback Command*) is Cleared. This bit is Cleared when the USB 3382 exits the *Loopback.Active* substate.	RO	No	0

Register 15-80. 230h Physical Layer Port/SuperSpeed USB Command (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^{2}C	Default
8	**Port 2 Loopback Command** 0 = Port 2 is not enabled to go to the *Loopback* Master state. 1 = Port 2 attempts to enter the *Loopback* state as a Loopback Master. If this bit is Set before the *Configuration* state is reached, the *Configuration.Linkwidth.Start* to *Loopback* path is used. If this bit is Set later, the *Recovery.Idle* to *Loopback* path is used.	RWS	Yes	0
9	**Port 2 Scrambler Disable Command** When Set, unconditionally disables the data scramblers on Lane 2, and causes the *Disable Scrambling* Training Control Bit to be Set in the transmitted Training Sets. If a serial EEPROM load Sets this bit, the scrambler is disabled in a *Configuration.Complete* substate. If software Sets this bit when the Link is in the Up state, hardware disables its scrambler without executing the Link Training protocol. This scrambler disable takes effect after the Link passes through *Configuration* again. The upstream/downstream device scrambler will not be disabled. 0 = Port 2's scrambler is enabled 1 = Port 2's scrambler is disabled	RWS	Yes	0
10	**Port 2 Compliance Receive Command** 0 = When Port 2 transmits TS1 Ordered-Sets, the *Compliance Receive* Training Control Bit within these Ordered-Sets is not Set during the *Polling.Active* nor *Loopback.Entry* substate 1 = When Port 2 transmits TS1 Ordered-Sets, the *Compliance Receive* Training Control Bit within these Ordered-Sets is Set during the *Polling.Active* or *Loopback.Entry* substate	RWS	Yes	0
11	**Port 2 Ready as Loopback Master** LTSSM established Loopback as a Master for Port 2. 0 = Port 2 is not in Loopback Master mode. 1 = Indicates that Port 2 has successfully transitioned to the *Loopback.Active* substate as a Loopback Master. The LTSSM remains in this substate, until bit 8 (*Port 2 Loopback Command*) is Cleared. This bit is Cleared when the USB 3382 exits the *Loopback.Active* substate.	RO	No	0

Register 15-80. 230h Physical Layer Port/SuperSpeed USB Command (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
12	**SuperSpeed USB Loopback Command** 0 = SuperSpeed USB is not enabled to go to the *Loopback* Master state. 1 = SuperSpeed USB attempts to enter the *Loopback* state as a Loopback Master. If this bit is Set before the *Configuration* state is reached, the *Configuration.Linkwidth.Start* to *Loopback* path is used. If this bit is Set later, the *Recovery.Idle* to *Loopback* path is used.	RWS	Yes	0
13	**SuperSpeed USB Scrambler Disable Command** When Set, unconditionally disables the data scramblers on the SuperSpeed USB, and causes the *Disable Scrambling* Training Control Bit to be Set in the transmitted Training Sets. If a serial EEPROM load Sets this bit, the scrambler is disabled in a *Configuration.Complete* substate. If software Sets this bit when the Link is in the Up state, hardware disables its scrambler without executing the Link Training protocol. This scrambler disable takes effect after the Link passes through *Configuration* again. The upstream/downstream device scrambler will not be disabled. 0 = SuperSpeed USB's scrambler is enabled 1 = SuperSpeed USB's scrambler is disabled	RWS	Yes	0
14	**SuperSpeed USB Compliance Receive Command** 0 = When the SuperSpeed USB transmits TS1 Ordered-Sets, the *Compliance Receive* Training Control Bit within these Ordered-Sets is not Set during the *Polling.Active* nor *Loopback.Entry* substate 1 = When the SuperSpeed USB transmits TS1 Ordered-Sets, the *Compliance Receive* Training Control Bit within these Ordered-Sets is Set during the *Polling.Active* or *Loopback.Entry* substate	RWS	Yes	0
15	**SuperSpeed USB Ready as Loopback Master** LTSSM established Loopback as a Master for the SuperSpeed USB. 0 = SuperSpeed USB is not in Loopback Master mode. 1 = Indicates that the SuperSpeed USB has successfully transitioned to the *Loopback.Active* substate as a Loopback Master. The LTSSM remains in this substate, until bit 12 (*SuperSpeed USB Loopback Command*) is Cleared. This bit is Cleared when the USB 3382 exits the *Loopback.Active* substate.	RO	No	0
31:16	***Reserved***	RsvdP	No	0000h

Register 15-81. 234h SKIP Ordered-Set Interval, Port and SuperSpeed USB Control (Port 0)

<table>
<tr><th>Bit(s)</th><th>Description</th><th>Type</th><th>Serial EEPROM and I^2C</th><th>Default</th></tr>
<tr><td colspan="5">The non-reserved upper 16 bits of this register (also referred to as the Port and SuperSpeed USB Control register) are used to disable or enable the LTSSM within individual Ports and the SuperSpeed USB. The bits are intended to be used in lieu of placing the Port and/or SuperSpeed USB into the Loopback.Active substate as a Loopback Master. These bits enable the test patterns to be transmitted, with or without a device attached at the far end. Recommended usage is as follows:
1. Set the Port and/or SuperSpeed USB’s Disable Port x/SuperSpeed USB and Port x/SuperSpeed USB Quiet bits.
• Setting the Port and/or SuperSpeed USB’s Disable Port x/SuperSpeed USB bit forces the Port and/or SuperSpeed USB into the Detect.Quiet substate. If no device is attached, it is not necessary to Set the Port and/or SuperSpeed USB’s Disable Port x/SuperSpeed USB bit.
• If 5.0 GT/s is needed, also Set the Port/SuperSpeed USB’s Test Pattern x Rate bit.
2. If Set, Clear the Port and/or SuperSpeed USB’s Disable Port x/SuperSpeed USB bit.
3. Load the UTP registers and enable UTP transmission, or just enable PRBS transmission.</td></tr>
<tr><td>11:0</td><td>SKIP Ordered-Set Interval
Specifies the SKIP Ordered-Set interval (in symbol times). When a value of 000h is written, SKIP Ordered-Set transmission is disabled.
000h = Disables SKIP Ordered-Set transmission
49Ch = Minimum interval (1,180 symbol times)
602h = Maximum interval (1,538 symbol times)
Note: A high value (such as FFFh) can cause the Link to fail.</td><td>RWS</td><td>Yes</td><td>49Ch</td></tr>
<tr><td>15:12</td><td>Reserved</td><td>RsvdP</td><td>No</td><td>0h</td></tr>
</table>

Register 15-81. 234h SKIP Ordered-Set Interval, Port and SuperSpeed USB Control (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
16	**Disable Port 0** 0 = Enables Link Training operation on Port 0. 1 = LTSSM remains in the *Detect.Quiet* substate on Port 0, if it is currently in, or returns to, that substate. Unconditionally disables Port 0. This is different from the Link Training and Status State Machine (LTSSM) *Disabled* state, in that Port 0 does not attempt to enter this state. If Port 0 is idle, it ceases attempting to detect a Receiver. If Port 0 is up, it immediately returns to the *Detect.Quiet* substate and remains there. No Electrical Idle Ordered-Set (EIOS) is sent, which could force any connected device to the *Recovery* state, and then to the LTSSM *Detect* state. Port 0 remains disabled until this bit is Cleared. While Port 0 is disabled, the SerDes that belong to Port 0 is (are) placed into the P1 SerDes Power state.	RWS	Yes	0
17	**Disable Port 1** 0 = Enables Link Training operation on Port 1. 1 = LTSSM remains in the *Detect.Quiet* substate on Port 1, if it is currently in, or returns to, that substate. Unconditionally disables Port 1. This is different from the LTSSM *Disabled* state, in that Port 1 does not attempt to enter this state. If Port 1 is idle, it ceases attempting to detect a Receiver. If Port 1 is up, it immediately returns to the *Detect.Quiet* substate and remains there. No EIOS is sent, which could force any connected device to the *Recovery* state, and then to the LTSSM *Detect* state. Port 1 remains disabled until this bit is Cleared. While Port 1 is disabled, SerDes 1 is placed into the P1 SerDes Power state.	RWS	Yes	0
18	**Disable Port 2** 0 = Enables Link Training operation on Port 2. 1 = LTSSM remains in the *Detect.Quiet* substate on Port 2, if it is currently in, or returns to, that substate. Unconditionally disables Port 2. This is different from the LTSSM *Disabled* state, in that Port 2 does not attempt to enter this state. If Port 2 is idle, it ceases attempting to detect a Receiver. If Port 2 is up, it immediately returns to the *Detect.Quiet* substate and remains there. No EIOS is sent, which could force any connected device to the *Recovery* state, and then to the LTSSM *Detect* state. Port 2 remains disabled until this bit is Cleared. While Port 2 is disabled, SerDes 2 is placed into the P1 SerDes Power state.	RWS	Yes	0
19	**Disable SuperSpeed USB** 0 = Enables Link Training operation on the SuperSpeed USB. 1 = LTSSM remains in the *Detect.Quiet* substate on the SuperSpeed USB, if it is currently in, or returns to, that substate. Unconditionally disables the SuperSpeed USB. This is different from the LTSSM *Disabled* state, in that the SuperSpeed USB does not attempt to enter this state. If the SuperSpeed USB is idle, it ceases attempting to detect a Receiver. If the SuperSpeed USB is up, it immediately returns to the *Detect.Quiet* substate and remains there. No EIOS is sent, which could force any connected device to the *Recovery* state, and then to the LTSSM *Detect* state. The SuperSpeed USB remains disabled until this bit is Cleared. While the SuperSpeed USB is disabled, it is placed into the P1 SerDes Power state.	RWS	Yes	0

Register 15-81. 234h SKIP Ordered-Set Interval, Port and SuperSpeed USB Control (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
20	**Port 0 Quiet** Once in the *Detect.Quiet* substate, Receiver termination is enabled and the Transmitters are placed into the P0 SerDes Power state. Port 0 can now transmit test patterns (PRBS or UTP), with or without an attached device and without being in the *Loopback.Active* substate. 0 = LTSSM is allowed to exit the *Detect.Quiet* substate in a normal manner (no effect on the LTSSM) 1 = LTSSM remains in the *Detect.Quiet* substate on Port 0 if it is currently in, or returns to, that substate ***Note:*** *Use this bit when it is necessary to transmit some data pattern, without first entering the Loopback.Active substate as a Loopback Master.*	RWS	Yes	0
21	**Port 1 Quiet** Once in the *Detect.Quiet* substate, Receiver termination is enabled and the Transmitters are placed into the P0 SerDes Power state. Port 1 can now transmit test patterns (PRBS or UTP), with or without an attached device and without being in the *Loopback.Active* substate. 0 = LTSSM is allowed to exit the *Detect.Quiet* substate in a normal manner (no effect on the LTSSM) 1 = LTSSM remains in the *Detect.Quiet* substate on Port 1 if it is currently in, or returns to, that substate ***Note:*** *Use this bit when it is necessary to transmit some data pattern, without first entering the Loopback.Active substate as a Loopback Master.*	RWS	Yes	0
22	**Port 2 Quiet** Once in the *Detect.Quiet* substate, Receiver termination is enabled and the Transmitters are placed into the P0 SerDes Power state. Port 2 can now transmit test patterns (PRBS or UTP), with or without an attached device and without being in the *Loopback.Active* substate. 0 = LTSSM is allowed to exit the *Detect.Quiet* substate in a normal manner (no effect on the LTSSM) 1 = LTSSM remains in the *Detect.Quiet* substate on Port 2 if it is currently in, or returns to, that substate ***Note:*** *Use this bit when it is necessary to transmit some data pattern, without first entering the Loopback.Active substate as a Loopback Master.*	RWS	Yes	0
23	**SuperSpeed USB Quiet** Once in the *Detect.Quiet* substate, Receiver termination is enabled and the Transmitters are placed into the P0 SerDes Power state. The SuperSpeed USB can now transmit test patterns (PRBS or UTP), with or without an attached device and without being in the *Loopback.Active* substate. 0 = LTSSM is allowed to exit the *Detect.Quiet* substate in a normal manner (no effect on the LTSSM) 1 = LTSSM remains in the *Detect.Quiet* substate on the SuperSpeed USB if it is currently in, or returns to, that substate ***Note:*** *Use this bit when it is necessary to transmit some data pattern, without first entering the Loopback.Active substate as a Loopback Master.*	RWS	Yes	0

Register 15-81. 234h SKIP Ordered-Set Interval, Port and SuperSpeed USB Control (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
24	**Port 0 Test Pattern *x* Rate** Port 0 transmits the selected test pattern (PRBS or UTP) at 5.0 GT/s, if bit 20 (*Port 0 Quiet*) is also Set (manual rate selection is enabled only when the *Quiet* bit is Set). 0 = Test pattern is transmitted at the Gen 1 Link rate (2.5 GT/s) 1 = Test pattern is transmitted at the Gen 2 Link (5.0 GT/s)	RWS	Yes	0
25	**Port 1 Test Pattern *x* Rate** Port 1 transmits the selected test pattern (PRBS or UTP) at 5.0 GT/s, if bit 21 (*Port 1 Quiet*) is also Set (manual rate selection is enabled only when the *Quiet* bit is Set). 0 = Test pattern is transmitted at a rate of 2.5 GT/s 1 = Test pattern is transmitted at a rate of 5.0 GT/s	RWS	Yes	0
26	**Port 2 Test Pattern *x* Rate** Port 2 transmits the selected test pattern (PRBS or UTP) at 5.0 GT/s, if bit 22 (*Port 2 Quiet*) is also Set (manual rate selection is enabled only when the *Quiet* bit is Set). 0 = Test pattern is transmitted at a rate of 2.5 GT/s 1 = Test pattern is transmitted at a rate of 5.0 GT/s	RWS	Yes	0
27	**SuperSpeed USB Test Pattern *x* Rate** The SuperSpeed USB transmits the selected test pattern (PRBS or UTP) at 5.0 GT/s, if bit 23 (*SuperSpeed USB Quiet*) is also Set (manual rate selection is enabled only when the *Quiet* bit is Set). 0 = Test pattern is transmitted at a rate of 2.5 GT/s 1 = Test pattern is transmitted at a rate of 5.0 GT/s	RWS	Yes	0

Register 15-81. 234h SKIP Ordered-Set Interval, Port and SuperSpeed USB Control (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
28	**Port 0 Bypass UTP Alignment Pattern** Must be Set if the following conditions exist: • Port 0 is a Loopback Master transmitting the User Test Pattern, and • Loopback Slave is in a different clock domain 0 = UTP Transmitter continuously transmits the alignment pattern until any UTP checker in either SerDes associated with Port 0 indicates that it has received the alignment pattern. The UTP Transmitter will then transmit one sync pattern, followed by the programmed user test pattern. 1 = Programmed user test pattern will be preceded by one alignment pattern (D3.2 D18.2 D13.2 D17.1) and one sync pattern (K28.5 D3.2 D18.2 D13.2).	RWS	Yes	0
29	**Port 1 Bypass UTP Alignment Pattern** Must be Set if the following conditions exist: • Port 1 is a Loopback Master transmitting the User Test Pattern, and • Loopback Slave is in a different clock domain 0 = UTP Transmitter continuously transmits the alignment pattern until any UTP checker in SerDes 1 indicates that it has received the alignment pattern. The UTP Transmitter will then transmit one sync pattern, followed by the programmed user test pattern. 1 = Programmed user test pattern will be preceded by one alignment pattern (D3.2 D18.2 D13.2 D17.1) and one sync pattern (K28.5 D3.2 D18.2 D13.2).	RWS	Yes	0
30	**Port 2 Bypass UTP Alignment Pattern** Must be Set if the following conditions exist: • Port 2 is a Loopback Master transmitting the User Test Pattern, and • Loopback Slave is in a different clock domain 0 = UTP Transmitter continuously transmits the alignment pattern until any UTP checker in SerDes 2 indicates that it has received the alignment pattern. The UTP Transmitter will then transmit one sync pattern, followed by the programmed user test pattern. 1 = Programmed user test pattern will be preceded by one alignment pattern (D3.2 D18.2 D13.2 D17.1) and one sync pattern (K28.5 D3.2 D18.2 D13.2).	RWS	Yes	0
31	**SuperSpeed USB Bypass UTP Alignment Pattern** Must be Set if the following conditions exist: • SuperSpeed USB is a Loopback Master transmitting the User Test Pattern, and • Loopback Slave is in a different clock domain 0 = UTP Transmitter continuously transmits the alignment pattern until any UTP checker in the SuperSpeed USB indicates that it has received the alignment pattern. The UTP Transmitter will then transmit one sync pattern, followed by the programmed user test pattern. 1 = Programmed user test pattern will be preceded by one alignment pattern (D3.2 D18.2 D13.2 D17.1) and one sync pattern (K28.5 D3.2 D18.2 D13.2).	RWS	Yes	0

Register 15-82. 238h SerDes Diagnostic Data (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
This register is used to retrieve Diagnostic Test results for SerDes[0-2] and the SuperSpeed USB.				
7:0	**UTP Expected Data** When User Test Pattern (UTP) is enabled, if an error is detected in the received UTP data, this field returns the expected data.	RO	No	00h
15:8	**UTP Actual Data** When UTP is enabled, if an error is detected in the received UTP data, this field returns the actual data that was received.	RO	No	00h
23:16	**UTP/PRBS Error Counter** Receiver Detected flags. Returns the quantity of errors detected by the UTP (bit 30 is Cleared) or PRBS (bit 30 is Set) Data Checkers. The Error Counter saturates at 255, and is Cleared when UTP/PRBS is disabled. **UTP Mode** To Clear the Counter, disable UTP mode by Clearing one or more of the **Physical Layer Test** register *SerDes x/SuperSpeed USB User Test Pattern Enable* bit(s) (Port 0, offset 228h[31:28]). **PRBS Mode** To Clear the Counter, disable PRBS mode by Clearing one or more of the **SerDes Test** register *SerDes x/SuperSpeed USB BIST Generator/ Checker Enable* bit(s) (Port 0, offset B88h[19:16]).	RO	No	00h
25:24	**SerDes Diagnostic Data Select** Used to select the SerDes[0-2] or the SuperSpeed USB to which the diagnostic data in this register pertains. Status selection code for the fields representing RO bits [30, 23:0] of this register. The binary code represents a status selection for Lane 0, 1, 2, or the SuperSpeed USB. The test results for physical device Lanes [0-2] or the SuperSpeed USB are selected with corresponding binary codes from 0-3, respectively. ***Note:*** *To obtain diagnostic data on all SerDes and the SuperSpeed USB, run a test, then cycle these bits.*	RW	Yes	00b
29:26	***Reserved***	RsvdP	No	0h
30	**PRBS Counter/-UTP Counter** 0 = Indicates that field [23:16] (*UTP/PRBS Error Counter*) is the UTP Error Counter 1 = Indicates that field [23:16] (*UTP/PRBS Error Counter*) is the PRBS Error Counter	RO	No	0
31	***Reserved***	RsvdP	No	0

Register 15-83. 248h Port Receiver Error Counters (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
7:0	**Port 0 Receiver Error Counter** When read, returns the quantity of Receiver errors detected by Port 0. The Error Counter saturates at 255. The Counter is Cleared with any Write to the corresponding byte in this register; otherwise, this field is RO.	RW1C	No	00h
15:8	**Port 1 Receiver Error Counter** When read, returns the quantity of Receiver errors detected by Port 1. The Error Counter saturates at 255. The Counter is Cleared with any Write to the corresponding byte in this register; otherwise, this field is RO.	RW1C	No	00h
23:16	**Port 2 Receiver Error Counter** When read, returns the quantity of Receiver errors detected by Port 2. The Error Counter saturates at 255. The Counter is Cleared with any Write to the corresponding byte in this register; otherwise, this field is RO.	RW1C	No	00h
31:24	**SuperSpeed USB Receiver Error Counter** When read, returns the quantity of Receiver errors detected by the SuperSpeed USB. The Error Counter saturates at 255. The Counter is Cleared with any Write to the corresponding byte in this register; otherwise, this field is RO.	RW1C	No	00h

Register 15-84. 24Ch Target Link Width (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
This register contains Controls and Status for Link width re-configuration on x2-capable Port 0, in the PHY partition. Link width fields in this register are provided, to enable software to direct Link width Up/Down configuration. If the Target Link width is not equal to the current Link width, the LTSSM transitions from *Recovery* to *Configuration*, then re-negotiates the Link width.				
1:0	**Port 0 Target Link Width** Provides the capability to allow software to cause Link retraining to a wider or narrower width than the current width, *such as* to conserve power when maximum bandwidth is not required. Devices advertise such support, by transmitting the appropriate value for the Data Rate Identifier symbol within TS2 Ordered-Sets. Written with the Target Link width for Port 0, to support Link Width Upconfiguration. The initial value of this field is the Port's Negotiated Link Width (All Ports and USB Controller, offset 78h[25:20]).	RWS	No	Defined by STRAP_PORTCFG input state, or programmed by serial EEPROM value for the **Port Configuration** register *Port Configuration* bit (Port 0, offset 574h[0])
6:2	***Reserved***	RsvdP	No	0-0h
7	**Port 0 Upconfigure Capability Received** Set during Link training, if Port 0 received an Upconfigure Capability notification from the connected device. 0 = Device connected to the Port does not indicate that it is capable of Link Width Upconfiguration. 1 = Device connected to the Port indicates that it is capable of Link Width Upconfiguration.	RO	No	0
31:8	***Reserved***	RsvdP	No	0000_00h

Register 15-85. 254h Physical Layer Additional Status (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
0	**Port 0 Loopback Master Entry Failed** 1 = Indicates that Port 0 failed to enter the *Loopback* state as a Loopback Master, and abandoned the attempt by returning the LTSSM to the *Detect* state ***Note:*** *If this bit and the **Physical Layer Port/SuperSpeed USB Command** register Port 0 Ready as Loopback Master bit (Port 0, offset 230h[3]) are both Set, the Loopback state was entered after the initial failure from the Configuration state.*	RW1C	Yes	0
1	**Port 1 Loopback Master Entry Failed** 1 = Indicates that Port 1 failed to enter the *Loopback* state as a Loopback Master, and abandoned the attempt by returning the LTSSM to the *Detect* state ***Note:*** *If this bit and the **Physical Layer Port/SuperSpeed USB Command** register Port 1 Ready as Loopback Master bit (Port 0, offset 230h[7]) are both Set, the Loopback state was entered after the initial failure from the Configuration state.*	RW1C	Yes	0
2	**Port 2 Loopback Master Entry Failed** 1 = Indicates that Port 2 failed to enter the *Loopback* state as a Loopback Master, and abandoned the attempt by returning the LTSSM to the *Detect* state ***Note:*** *If this bit and the **Physical Layer Port/SuperSpeed USB Command** register Port 2 Ready as Loopback Master bit (Port 0, offset 230h[11]) are both Set, the Loopback state was entered after the initial failure from the Configuration state.*	RW1C	Yes	0
3	**SuperSpeed USB Loopback Master Entry Failed** 1 = Indicates that the SuperSpeed USB failed to enter the *Loopback* state as a Loopback Master, and abandoned the attempt by returning the LTSSM to the *Detect* state ***Note:*** *If this bit and the **Physical Layer Port/SuperSpeed USB Command** register SuperSpeed USB Ready as Loopback Master bit (Port 0, offset 230h[15]) are both Set, the Loopback state was entered after the initial failure from the Configuration state.*	RW1C	Yes	0
7:4	**PhyStatus** PIPE interface PhyStatus.	RO	No	PCFG
14:8	**Received Modified Compliance Error Counter**	RO	No	0-0h
15	**Received Modified Compliance Pattern Lock**	RO	No	0

Register 15-85. 254h Physical Layer Additional Status (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
16	**Port 0 External Loopback Enable** 1 = Allows Port 0 to reach Link Up status, when receiving its own Training Sets during Link training. It is necessary to Set this bit when the Port 0 Receivers are directly connected, externally, to its Transmitters.	RW	Yes	0
17	**Port 1 External Loopback Enable** 1 = Allows Port 1 to reach Link Up status, when receiving its own Training Sets during Link training. It is necessary to Set this bit when the Port 1 Receivers are directly connected, externally, to its Transmitters.	RW	Yes	0
18	**Port 2 External Loopback Enable** 1 = Allows Port 2 to reach Link Up status, when receiving its own Training Sets during Link training. It is necessary to Set this bit when the Port 2 Receivers are directly connected, externally, to its Transmitters.	RW	Yes	0
19	**SuperSpeed USB External Loopback Enable** 1 = Allows the SuperSpeed USB to reach Link Up status, when receiving its own Training Sets during Link training. It is necessary to Set this bit when the SuperSpeed USB Receivers are directly connected, externally, to its Transmitters.	RW	Yes	0
23:20	**2nd Receiver Detect Disable**	RWS	Yes	0h
25:24	**Received Modified Compliance Lane Select**	RW	Yes	00b
27:26	***Reserved***	RsvdP	No	00b
28	***Factory Test Only***	RW	Yes	0
31:29	***Reserved***	RsvdP	No	000b

Register 15-86. 258h PRBS Control/Status (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
0	**PRBS Pattern Sync Status Device Lane 0** Diagnoses broken Loopback wiring, because no errors can be detected and counted until pattern sync is detected. Sync achieved is Active-High. 0 = Causes the PRBS pattern generator, when enabled, to output the PRBS7 sequence 1 = Indicates that either Lane associated with Port 0's PRBS Data Checker has synchronized to the incoming PRBS data pattern – the Receiver has acquired its first match on two sequentially received words	RO	No	0
1	**PRBS Pattern Sync Status Device Lane 1** Diagnoses broken Loopback wiring, because no errors can be detected and counted until pattern sync is detected. Sync achieved is Active-High. 0 = Causes the PRBS pattern generator, when enabled, to output the PRBS7 sequence 1 = Indicates that Lane 1's PRBS Data Checker has synchronized to the incoming PRBS data pattern – the Receiver has acquired its first match on two sequentially received words	RO	No	0
2	**PRBS Pattern Sync Status Device Lane 2** Diagnoses broken Loopback wiring, because no errors can be detected and counted until pattern sync is detected. Sync achieved is Active-High. 0 = Causes the PRBS pattern generator, when enabled, to output the PRBS7 sequence 1 = Indicates that Lane 2's PRBS Data Checker has synchronized to the incoming PRBS data pattern – the Receiver has acquired its first match on two sequentially received words	RO	No	0
3	**PRBS Pattern Sync Status Device SuperSpeed USB** Diagnoses broken Loopback wiring, because no errors can be detected and counted until pattern sync is detected. Sync achieved is Active-High. 0 = Causes the PRBS pattern generator, when enabled, to output the PRBS7 sequence 1 = Indicates that the SuperSpeed USB's PRBS Data Checker has synchronized to the incoming PRBS data pattern – the Receiver has acquired its first match on two sequentially received words	RO	No	0
15:4	***Reserved***	RsvdP	No	000h
16	**PRBS Pattern Invert Enable** 1 = Causes the PRBS pattern generator to output the one's complement of the PRBS7 sequence	RW	Yes	0
31:17	***Reserved***	RsvdP	No	0-0h

15.14.3 Device-Specific Registers – Serial EEPROM (Offsets 260h – 26Ch)

This section details the Device-Specific Serial EEPROM registers. Table 15-19 defines the register map.

Table 15-19. Device-Specific Serial EEPROM Register Map (Port 0)

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Status Data from Serial EEPROM	Serial EEPROM Status	Serial EEPROM Control		260h
Serial EEPROM Buffer				264h
Serial EEPROM Clock Frequency				268h
Reserved			Serial EEPROM 3rd Address Byte	26Ch

Register 15-87. 260h Serial EEPROM Status and Control (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^{2}C	Default
	Serial EEPROM Control			
12:0	**EepBlkAddr** Serial EEPROM Block Address for 32 KB.	RW	Yes	000h
15:13	**EepCmd[2:0]** Commands to the Serial EEPROM Controller. 000b = ***Reserved*** 001b = Data from bits [31:24] (**Status Data from Serial EEPROM** register) is written to the serial EEPROM's internal **Status** register 010b = Write four bytes of data from the EepBuf into the memory location pointed to by field [12:0] (*EepBlkAddr*) 011b = Read four bytes of data from the memory location pointed to by field [12:0] (*EepBlkAddr*) into the EepBuf 100b = Reset Write Enable latch 101b = Data from the serial EEPROM's internal **Status** register is written to bits [31:24] (**Status Data from Serial EEPROM** register) 110b = Set Write Enable latch 111b = ***Reserved*** ***Note:*** *For value of 001b, only bits [31, 27:26] can be written into the serial EEPROM's internal **Status** register.*	RW	Yes	000b

Register 15-87. 260h Serial EEPROM Status and Control (Port 0) *(Cont.)*

<table>
<tr><th>Bit(s)</th><th colspan="2">Description</th><th>Type</th><th>Serial EEPROM and I²C</th><th>Default</th></tr>
<tr><td colspan="6">Serial EEPROM Status</td></tr>
<tr><td>17:16</td><td colspan="2">EepPrsnt[1:0]
Serial EEPROM Present status.
00b = Not present
01b = Serial EEPROM is present – Validation Signature is verified
10b = Reserved
11b = Serial EEPROM is present – Validation Signature is not verified</td><td>HwInit</td><td>No</td><td>00b</td></tr>
<tr><td>18</td><td colspan="2">EepCmdStatus
Serial EEPROM Command status.
0 = Serial EEPROM Command is complete
1 = Serial EEPROM Command is not complete</td><td>RO</td><td>No</td><td>0</td></tr>
<tr><td>19</td><td colspan="2">Reserved</td><td>RsvdP</td><td>No</td><td>0</td></tr>
<tr><td>20</td><td colspan="2">EepBlkAddr Upper Bit
Serial EEPROM Block Address upper bit 13. Extends the serial EEPROM to 64 KB.</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td>21</td><td colspan="2">EepAddrWidth Override
0 = Field [23:22] (EepAddrWidth) is RO
1 = Field [23:22] (EepAddrWidth) is software-writable</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td rowspan="2">23:22</td><td rowspan="2">EepAddrWidth
Serial EEPROM Address width. If the addressing width cannot be determined, 00b is returned. A non-zero value is reported only if the validation signature (5Ah) is successfully read from the first serial EEPROM location.
This field is usually HwInit; however, it is RW if bit 21 (EepAddrWidth Override) is Set.
00b = Undetermined
01b = 1 byte
10b = 2 bytes
11b = 3 bytes</td><td>Bit 21 = 0</td><td>HwInit</td><td>No</td><td>00b</td></tr>
<tr><td>Bit 21 = 1</td><td>RW</td><td>No</td><td>00b</td></tr>
</table>

Register 15-87. 260h Serial EEPROM Status and Control (Port 0) *(Cont.)*

<table>
<tr><th>Bit(s)</th><th>Description</th><th>Type</th><th>Serial EEPROM and I²C</th><th>Default</th></tr>
<tr><td colspan="5">Status Data from Serial EEPROM[a]</td></tr>
<tr><td>24</td><td>EepRdy
Serial EEPROM RDY#.
0 = Serial EEPROM is ready to transmit data
1 = Write cycle is in progress</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td>25</td><td>EepWen
Serial EEPROM Write enable.
0 = Serial EEPROM Write is disabled
1 = Serial EEPROM Write is enabled</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td>27:26</td><td>EepBp[1:0]
Serial EEPROM Block-Write Protect bits. Block Protection options protect the top ¼, top ½, or the entire serial EEPROM. USB 3382 Configuration data is stored in the lower addresses; therefore, when using Block Protection, the entire serial EEPROM should be protected with BP[1:0]=11b.
<table>
<tr><th rowspan="2">BP[1:0]</th><th rowspan="2">Level</th><th colspan="4">Array Addresses Protected, by Device Size</th></tr>
<tr><th>8 KB</th><th>16 KB</th><th>32 KB</th><th>64 KB</th></tr>
<tr><td>00b</td><td>0</td><td>None</td><td>None</td><td>None</td><td>None</td></tr>
<tr><td>01b</td><td>1
(top ¼)</td><td>1800h – 1FFFh</td><td>3000h – 3FFFh</td><td>6000h – 7FFFh</td><td>–</td></tr>
<tr><td>10b</td><td>2
(top ½)</td><td>1000h – 1FFFh</td><td>2000h – 3FFFh</td><td>4000h – 7FFFh</td><td>–</td></tr>
<tr><td>11b</td><td>3
(All)</td><td>0000h – 1FFFh</td><td>0000h – 3FFFh</td><td>0000h – 7FFFh</td><td>–</td></tr>
</table></td><td>RW</td><td>Yes</td><td>00b</td></tr>
</table>

Register 15-87. 260h Serial EEPROM Status and Control (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
30:28	**EepWrStatus** Serial EEPROM Write status. Value is 000b when the serial EEPROM is not in an internal Write cycle. ***Note:*** *The definition of this field varies among serial EEPROM manufacturers. Reads of the serial EEPROM's internal* ***Status*** *register can return a value of 000b or 111b, depending upon the serial EEPROM that is used.*	RO	No	000b
31	**EepWpen** Serial EEPROM Write Protect enable. Overrides the internal serial EEPROM Write Protect WP# input and enables/disables Writes to the **Serial EEPROM Status** register (bits [23:16] of this register): • When WP#=H or this bit is Cleared, and bit 25 (*EepWen*) is Set, the **Serial EEPROM Status** register is writable • When WP#=L and this bit is Set, or bit 25 (*EepWen*) is Cleared, the **Serial EEPROM Status** register is write-protected ***Notes:*** *If the internal serial EEPROM Write Protect WP# input is Low, after software Sets the EepWen bit to write-protect the* ***Serial EEPROM Status*** *register, the EepWen value cannot be Cleared, nor can the EepBp[1:0] field be Cleared to disable Block Protection, until the WP# input is High.* *This bit is* ***not*** *implemented in certain serial EEPROMs. Refer to the serial EEPROM manufacturer's data sheet.*	RW	Yes	0

a. Within the serial EEPROM's internal ***Status*** *register, only bits [31, 27:26] can be written.*

Register 15-88. 264h Serial EEPROM Buffer (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
31:0	**EepBuf** Serial EEPROM RW buffer. Read/Write command to the **Serial EEPROM Control** register (Port 0, offset 260h) results in a 4-byte Read/Write to/from the serial EEPROM device.	RW	Yes	0000_0000h

Register 15-89. 268h Serial EEPROM Clock Frequency (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
2:0	**EepFreq[2:0]** Serial EEPROM clock (EE_CLK/EE_SK) frequency control. 000b = 1 MHz (default) 001b = 1.98 MHz 010b = 5 MHz 011b = 9.62 MHz 100b = 12.5 MHz 101b = 15.6 MHz 110b = 17.86 MHz 111b = ***Reserved***	RW	Yes	000b
7:3	***Reserved***	RsvdP	No	0-0h
10:8	**EepCsStHld[2:0]** CS to SCLK setup and hold timing to the serial EEPROM, between EE_CS# active and EE_CLK/EE_SK active, and between EE_CLK/EE_SK inactive and EE_CS# inactive. Time increases in ½ EE_CLK/EE_SK Clock cycle increments, from a minimum of ½, to a maximum of 4. 000b = ½ EE_CLK/EE_SK clocks (minimum time) 001b = 1 EE_CLK/EE_SK clock (+ ½ clock) 010b = 1 ½ EE_CLK/EE_SK clocks 011b = 2 EE_CLK/EE_SK clocks 100b = 2 ½ EE_CLK/EE_SK clocks 101b = 3 EE_CLK/EE_SK clocks 110b = 3 ½ EE_CLK/EE_SK clocks 111b = 4 EE_CLK/EE_SK clocks (maximum time)	RW	Yes	010b
31:11	***Reserved***	RsvdP	No	0-0h

Register 15-90. 26Ch Serial EEPROM 3rd Address Byte (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
7:0	**Serial EEPROM 3rd Address Byte**	RW	Yes	00h
31:8	***Reserved***	RsvdP	No	0000_00h

15.14.4 Device-Specific Registers – I²C and SMBus Slave Interfaces (Offsets 290h – 2C4h)

This section details the Device-Specific I²C Slave and SMBus Interface registers. Table 15-20 defines the register map.

The I²C and SMBus Slave Interfaces are described, in detail, in Chapter 12, "I²C/SMBus Slave Interface Operation."

Table 15-20. Device-Specific I²C and SMBus Slave Interfaces Register Map (Port 0)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Offset
Factory Test Only	290h
I²C Configuration	294h
Factory Test Only	298h – 2A8h
SMBus Configuration	2ACh
Factory Test Only/Reserved	2B0h – 2C4h

Register 15-91. 294h I²C Configuration (Port 0)

<table>
<tr><th>Bit(s)</th><th>Description</th><th>Type</th><th>Serial EEPROM and I²C</th><th colspan="2">Default</th></tr>
<tr><td>2:0</td><td rowspan="2">Slave Address
Bits [6:0] comprise the I²C Slave/SMBus Device address, 5Fh. The value is determined by bits [2:0] (which reflect the I2C_ADDR[2:0] input states, and default to 111b, by virtue of weak internal pull-up resistors), combined with the value of bits [6:3] (which default to 1011b).
When I2C_ADDR2=H, Address Resolution Protocol (ARP) is disabled and bit 2 defaults to a value of 1.
The following table lists the pin and register bit values associated with the I²C Slave and SMBus Device addresses. Other values can be programmed as well, by serial EEPROM or Memory Write.
<table>
<tr><th>I2C_ADDR[2:0] Values</th><th>Offset 294h[2:0] Value</th><th>Offset 294h[6:3] Value</th><th>I²C Slave Address</th><th>SMBus Device Address</th></tr>
<tr><td>000</td><td>000b</td><td rowspan="8">1011b</td><td>1011_000b</td><td>ARP</td></tr>
<tr><td>001</td><td>001b</td><td>1011_001b</td><td>ARP</td></tr>
<tr><td>010</td><td>010b</td><td>1011_010b</td><td>ARP</td></tr>
<tr><td>011</td><td>011b</td><td>1011_011b</td><td>ARP</td></tr>
<tr><td>100</td><td>100b</td><td>1011_100b</td><td>1011_100b</td></tr>
<tr><td>101</td><td>101b</td><td>1011_101b</td><td>1011_101b</td></tr>
<tr><td>110</td><td>110b</td><td>1011_110b</td><td>1011_110b</td></tr>
<tr><td>111</td><td>111b</td><td>1011_111b</td><td>1011_111b</td></tr>
</table>
Note: The I²C Slave/SMBus Device address must not be changed by an I²C/SMBus Write command.</td><td>HwInit</td><td>Yes</td><td>111b</td><td rowspan="2">5Fh</td></tr>
<tr><td>6:3</td><td>RWS</td><td>Yes</td><td>1011b</td></tr>
<tr><td>9:7</td><td>Reserved</td><td>RsvdP</td><td>No</td><td colspan="2">000b</td></tr>
<tr><td>10</td><td>Factory Test Only</td><td>RWS</td><td>Yes</td><td colspan="2">0</td></tr>
<tr><td>31:11</td><td>Reserved</td><td>RsvdP</td><td>No</td><td colspan="2">0000_00h</td></tr>
</table>

Register 15-92. 2ACh SMBus Configuration (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
0	**SMBus Enable** Initially loaded from the STRAP_SMBUS_EN# input state. The value can later be changed by serial EEPROM and/or Configuration Space register Read Write. 0 = Disables SMBus for device configuration (I^2C mode is enabled) 1 = Enables SMBus for device configuration (SMBus mode is enabled)	RWS	Yes	0 (STRAP_SMBUS_EN#=H) 1 (STRAP_SMBUS_EN#=L)
7:1	**SMBus Device Address** Defined by the Address Resolution Protocol (ARP) Master, if ARP is enabled. Defaults to 38h if ARP is disabled (I2C_ADDR2=H), with Address bit [1:0] values loaded from the I2C_ADDR[1:0] inputs.	RWS	Yes	00h (I2C_ADDR2=L) 38h (I2C_ADDR2=H)
8	**ARP Disable** 0 = Device under test is able to respond to ARP commands 1 = Device under test is unable to respond to ARP commands	RWS	Yes	0 (I2C_ADDR2=L) 1 (I2C_ADDR2=H)
9	**PEC Check Disable** 0 = Enable PEC checking on all packets 1 = Disables Packet Error Checks (PECs) checking on all packets; packets with the wrong PECs are accepted	RWS	Yes	0
10	**AV Flag** *Address Valid* (*AV*) flag. Set, by default, when ARP is disabled (I2C_ADDR2=H).	RWS	Yes	0 (I2C_ADDR2=L) 1 (I2C_ADDR2=H)
11	**AR Flag** *Address Resolved* (*AR*) flag.	RWS	Yes	0
13:12	**UDID Address Type** Unique Device Identifier (UDID) Address type. 00b = I2C_ADDR2=H (ARP is disabled) 10b = I2C_ADDR2=L (ARP is enabled) All other encodings are ***reserved***.	RWS	Yes	00b (I2C_ADDR2=H) 10b (I2C_ADDR2=L)
14	**UDID PEC Support** 1 = Sets the *PEC Support* bit in the UDID	RWS	Yes	1
15	**SMBus Parameter Re-Load** Set this bit if bits [10, 8, or 7:1] (*AV Flag*, *ARP Disable*, or *SMBus Device Address*, respectively) are changed after a serial EEPROM load. Effective only when bit 28 (*SMBus Command In-Progress*) is Cleared.	ROS	No	0

Register 15-92. 2ACh SMBus Configuration (Port 0) *(Cont.)*

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
23:16	**UDID Vendor-Specific ID** Sets the MSB of the UDID Vendor-Specific ID. Bits [23:20] of this field are programmed by the I2C_ADDR[1:0] inputs. The four combinations provide the following ID values: 00b = 7000_0000h 01b = B000_0000h 10b = D000_0000h 11b = E000_0000h	RWS	Yes	Defined by I2C_ADDR[1:0] input states
26:24	**UDID Version** UDID version defined for the *SMBus v2.0*.	RWS	Yes	001b
27	***Factory Test Only***	RWS	Yes	0
28	**SMBus Command In-Progress** 0 = SMBus state machine is idle 1 = SMBus state machine is active (not idle)	ROS	No	0
29	**PEC Check Failed** 0 = PEC check successfully completed when receiving a packet 1 = PEC check failed when receiving a packet	RW1CS	No	0
30	**Unsupported SMBus Command** 0 = Command received from SMBus is a supported command 1 = Command received from SMBus is an unsupported command	RW1CS	No	0
31	***Factory Test Only***	RW1CS	No	0

15.15 Device-Specific Registers (Offsets 530h – B88h)

This section details the Device-Specific registers located at offsets 530h through B88h. Device-Specific registers are unique to the USB 3382 and not referenced in the *PCI Express Base r2.1*. Table 15-21 defines the register map.

Other Device-Specific registers are detailed in Section 15.14, "Device-Specific Registers (Offsets 1C0h – 444h)."

Note:* *It is recommended that these registers not be changed from their default values.

Table 15-21. Device-Specific Register Map (Offsets 530h – B88h)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16		15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Factory Test Only/Reserved			530h – 570h
Device-Specific Registers – Port Configuration (Offset 574h)			574h
Factory Test Only/Reserved			578h – 65Ch
Device-Specific Registers – Negotiated Link Width (Offsets 660h – 67Ch)			660h … 67Ch
Reserved			680h – 94Ch
Next Capability Offset 2 (000h)	1h	PCI Express Extended Capability ID 2 (000Bh)	950h
Device-Specific Registers – Vendor-Specific Extended Capability 2 (Offsets 950h – 95Ch)			… 95Ch
Factory Test Only/Reserved			960h – 9ECh
Device-Specific Registers – Ingress Credit Handler Control and Status (Offsets 9F0h – 9FCh)			9F0h … 9FCh
Device-Specific Registers – Ingress Credit Handler Threshold (Offsets A00h – A2Ch)			A00h … A2Ch
Factory Test Only/Reserved			A30h – B7Ch
Device-Specific Registers – Physical Layer (Offsets B80h – B88h)			B80h … B88h

15.15.1 Device-Specific Registers – Port Configuration (Offset 574h)

This section details the Device-Specific Port Configuration register. Table 15-22 defines the register map.

Table 15-22. Device-Specific Port Configuration Register Map (Port 0)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Port Configuration	574h

Register 15-93. 574h Port Configuration (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Notes: *The Port configurations are listed in Table 15-12.* *Port 2 is the virtual PCI-to-PCI bridge, and is always enabled. It is not affected by the value of this register.*				
0	**Port Configuration** Port Configuration Link width, per Ports 0 and 1. The serial EEPROM bit values always override the STRAP_PORTCFG input (if the serial EEPROM values are loaded; refer to Table 15-12). This register is reset only by a Fundamental Reset (PEX_PERST# assertion). 0 = x1x1 (Ports 0 and 1; each Port operates at Gen 1 or Gen 2 Link rate) 1 = x2 (Port 0, Gen 1 Link rate only)	RO	Yes	Defined by STRAP_PORTCFG input state
31:1	***Reserved***	RsvdP	No	0-0h

15.15.2 Device-Specific Registers – Negotiated Link Width (Offsets 660h – 67Ch)

This section details the Device-Specific Negotiated Link Width register. Table 15-23 defines the register map.

Table 15-23. Device-Specific Negotiated Link Width Register Map (Port 0)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	Offset
Factory Test Only		660h – 668h
Reserved	Negotiated Link Width for Ports 0, 1, 2, and SuperSpeed USB	66Ch
Reserved		670h
Factory Test Only		674h – 67Ch

Register 15-94. 66Ch Negotiated Link Width for Ports 0, 1, 2, and SuperSpeed USB (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Note: *The downstream Ports and SuperSpeed USB are always x1 Link width.*				
0	**Negotiated Link Width for Port 0** 0 = x1 1 = x2	RO	No	Defined by STRAP_PORTCFG input state
1	**Link Speed for Port 0** 0 = Negotiated Link SerDes speed is 2.5 GT/s 1 = Negotiated Link SerDes speed is 5.0 GT/s	RO	No	0
2	**Negotiated Link Width for Port 1** 0 = x1 1 = ***Reserved***	RO	No	0
3	**Link Speed for Port 1** 0 = Negotiated Link SerDes speed is 2.5 GT/s 1 = Negotiated Link SerDes speed is 5.0 GT/s	RO	No	0
4	**Negotiated Link Width for Port 2** 0 = x1 1 = ***Reserved***	RO	No	0
5	**Valid Negotiated Link Width for Port 2** 0 = Negotiated Link SerDes speed is 2.5 GT/s 1 = Negotiated Link SerDes speed is 5.0 GT/s	RO	No	0
6	**Negotiated Link Width for SuperSpeed USB** ***Reserved***	RsvdP	No	0
7	**Valid Negotiated Link Speed for SuperSpeed USB** ***Reserved***	RsvdP	No	1
31:8	***Reserved***	RsvdP	No	0000_00h

15.15.3 Device-Specific Registers – Vendor-Specific Extended Capability 2 (Offsets 950h – 95Ch)

This section details the Device-Specific, Vendor-Specific Extended Capability 2 registers. Table 15-24 defines the register map.

Table 15-24. Device-Specific, Vendor-Specific Extended Capability 2 Register Map (All Ports and USB Controller)

31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Next Capability Offset 2 (000h)	Capability Version 2 (1h)	PCI Express Extended Capability ID 2 (000Bh)		950h
Vendor-Specific Header 2				954h
Hardwired Device ID		Hardwired Vendor ID		958h
Reserved			Hardwired Revision ID	95Ch

Register 15-95. 950h Vendor-Specific Extended Capability 2 (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
15:0	**PCI Express Extended Capability ID 2** Program to 000Bh, indicating that the Capability structure is the **Vendor-Specific Extended Capability** structure.	RO	Yes	000Bh
19:16	**Capability Version 2**	RO	Yes	1h
31:20	**Next Capability Offset 2** 000h = This extended capability is the last capability in the USB 3382 Extended Capabilities list	RO	Yes	000h

Register 15-96. 954h Vendor-Specific Header 2 (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
15:0	**Vendor-Specific ID 2** ID Number of this Extended Capability structure.	RO	Yes	0001h
19:16	**Vendor-Specific Rev 2** Version Number of this structure.	RO	Yes	0h
31:20	**Vendor-Specific Length 2** Quantity of bytes in the entire structure.	RO	Yes	028h

Register 15-97. 958h PLX Hardwired Configuration ID (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
15:0	**Hardwired Vendor ID** Always returns the PLX PCI-SIG-assigned Vendor ID value, 10B5h.	RO	No	10B5h
31:16	**Hardwired Device ID** Always returns the USB 3382 default Device ID value, 3382h.	RO	No	3382h

Register 15-98. 95Ch PLX Hardwired Revision ID (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
7:0	**Hardwired Revision ID** Always returns the USB 3382 default PCI Revision ID value, AAh or ABh.	RO	No	Current Rev # (AAh or ABh)
31:8	***Reserved***	RsvdP	No	0000_00h

15.15.4 Device-Specific Registers – Ingress Credit Handler Control and Status (Offsets 9F0h – 9FCh)

This section details the Device-Specific Ingress Credit Handler (INCH) Control and Status registers. Table 15-25 defines the register map.

Table 15-25. Device-Specific INCH Control and Status Register Map (Port 0)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Offset
INCH Status Control for Ports 0 and 1	9F0h
INCH Status Read for Ports 0 and 1	9F4h
INCH Status Control for Port 2 and SuperSpeed USB	9F8h
INCH Status Read for Port 2 and SuperSpeed USB	9FCh

Register 15-99. 9F0h INCH Status Control for Ports 0 and 1 (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
3:0	**Credit Available Select** Write the value here to specify which of the twelve Credit Available registers to read. This value is used as input to the **INCH Status Read for Ports 0 and 1** register *Read Credit Available Value* field (Port 0, offset 9F4h[19:0]). 0h = INCH Threshold Port 0 VC0 Posted credits available 1h = INCH Threshold Port 0 VC0 Non-Posted credits available 2h = INCH Threshold Port 0 VC0 Completion credits available 3h = INCH Threshold Port 1 VC0 Posted credits available 4h = INCH Threshold Port 1 VC0 Non-Posted credits available 5h = INCH Threshold Port 1 VC0 Completion credits available 8h = **INCH Threshold Port 0 VC0 Posted** register (Port 0, offset A00h) 9h = **INCH Threshold Port 0 VC0 Non-Posted** register (Port 0, offset A04h) Ah = **INCH Threshold Port 0 VC0 Completion** register (Port 0, offset A08h) Bh = **INCH Threshold Port 1 VC0 Posted** register (Port 0, offset A0Ch) Ch = **INCH Threshold Port 1 VC0 Non-Posted** register (Port 0, offset A10h) Dh = **INCH Threshold Port 1 VC0 Completion** register (Port 0, offset A14h) All other encodings are ***reserved***.	RWS	Yes	0h
23:4	***Reserved***	RsvdP	No	0h

Register 15-99. 9F0h INCH Status Control for Ports 0 and 1 (Port 0) *(Cont.)*

<table>
<tr><th>Bit(s)</th><th colspan="2">Description</th><th>Type</th><th>Serial EEPROM and I²C</th><th>Default</th></tr>
<tr><td>24</td><td colspan="2">FIFO to Port Map
Determines whether USB device Endpoint 0 (EP 0) is written by Port 0 or 1.
0 = EP_n is written by Port 0
1 = EP_n is written by Port 1</td><td>RWS</td><td>Yes</td><td>0</td></tr>
<tr><td>27:25</td><td colspan="2">Reserved</td><td>RsvdP</td><td>No</td><td>000b</td></tr>
<tr><td rowspan="6">31:28</td><td colspan="2">INCH FIFO OFF
Determines whether General-Purpose Endpoint x (GPEPx) is included in the level check. Bits [28:31] map to GPEP0 through GPEP3, respectively.
Note: In Root Complex mode, by default, the GPEP1 and GPEP3 IN FIFOs are not included in the credit calculations for Root Complex mode. (Refer to Section 8.5.1.1, “PCI Express Credits.”
0 = FIFO is included in the level check
1 = FIFO is not included in level check</td><td rowspan="6">RWS</td><td rowspan="6">Yes</td><td rowspan="6">Adapter mode:
0000b
Root Complex mode:
1010b</td></tr>
<tr><td>Bit</td><td>General-Purpose Endpoint</td></tr>
<tr><td>28</td><td>GPEP0</td></tr>
<tr><td>29</td><td>GPEP1</td></tr>
<tr><td>30</td><td>GPEP2</td></tr>
<tr><td>31</td><td>GPEP3</td></tr>
</table>

Register 15-100. 9F4h INCH Status Read for Ports 0 and 1 (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
19:0	**Read Credit Available Value** Read register selected in the **INCH Status Control for Ports 0 and 1** register *Credit Available Select* field (Port 0, offset 9F0h[3:0]). The value returns the selected Credit Available register.	RWS	Yes	0_0000h
31:20	***Reserved***	RsvdP	No	000h

Register 15-101. 9F8h INCH Status Control for Port 2 and SuperSpeed USB (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
3:0	**Credit Available Select** Write the value here to specify which of the 12 Credit Available registers to read. This value is used as input to the **INCH Status Read for Port 2 and SuperSpeed USB** register *Read Credit Available Value* field (Port 0, offset 9FCh[19:0]). 0h = INCH Threshold Port 2 VC0 Posted credits available (default) 1h = INCH Threshold Port 2 VC0 Non-Posted credits available 2h = INCH Threshold Port 2 VC0 Completion credits available 3h = INCH Threshold SuperSpeed USB VC0 Posted credits available 4h = INCH Threshold SuperSpeed USB VC0 Non-Posted credits available 5h = INCH Threshold SuperSpeed USB VC0 Completion credits available 8h = **INCH Threshold Port 2 VC0 Posted** register (Port 0, offset A18h) 9h = **INCH Threshold Port 2 VC0 Non-Posted** register (Port 0, offset A1Ch) Ah = **INCH Threshold Port 2 VC0 Completion** register (Port 0, offset A20h) Bh = **INCH Threshold SuperSpeed USB VC0 Posted** register (Port 0, offset A24h) Ch = **INCH Threshold SuperSpeed USB VC0 Non-Posted** register (Port 0, offset A28h) Dh = **INCH Threshold SuperSpeed USB VC0 Completion** register (Port 0, offset A2Ch) All other encodings are ***reserved***.	RWS	Yes	0h
25:4	***Reserved***	RsvdP	No	0-0h
29:26	**INCH FIFO OFF** Determines whether General-Purpose Endpoint *x* (GPEP*x*) is included in the level check. The bits map to GPEP0 through GPEP3, as follows: • Bit 26 maps to GPEP0 • Bit 27 maps to GPEP1 • Bit 28 maps to GPEP2 • Bit 29 maps to GPEP3 0 = FIFO is included in the level check 1 = FIFO is not included in level check	RWS	Yes	0h
31:30	***Reserved***	RsvdP	No	00b

Register 15-102. 9FCh INCH Status Read for Port 2 and SuperSpeed USB (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
19:0	**Read Credit Available Value** Read register selected in the **INCH Status Control for Port 2 and SuperSpeed USB** register *Credit Available Select* field (Port 0, offset 9F8h[3:0]). The value returns the selected Credit Available register.	RWS	Yes	0_0000h
31:20	***Reserved***	RsvdP	No	000h

15.15.5 Device-Specific Registers – Ingress Credit Handler Threshold (Offsets A00h – A2Ch)

This section details the Device-Specific Ingress Credit Handler (INCH) Threshold registers. **Changing credit values from default register values must be done carefully; otherwise, the USB 3382 will not properly function.** Table 15-26 defines the register map.

Table 15-26. Device-Specific INCH Threshold Register Map (Port 0)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
INCH Threshold Port 0 VC0 Posted		A00h
Factory Test Only/Reserved	INCH Threshold Port 0 VC0 Non-Posted	A04h
INCH Threshold Port 0 VC0 Completion		A08h
INCH Threshold Port 1 VC0 Posted		A0Ch
Factory Test Only/Reserved	INCH Threshold Port 1 VC0 Non-Posted	A10h
INCH Threshold Port 1 VC0 Completion		A14h
INCH Threshold Port 2 VC0 Posted		A18h
Factory Test Only/Reserved	INCH Threshold Port 2 VC0 Non-Posted	A1Ch
INCH Threshold Port 2 VC0 Completion		A20h
INCH Threshold SuperSpeed USB VC0 Posted		A24h
Factory Test Only/Reserved	INCH Threshold SuperSpeed USB VC0 Non-Posted	A28h
INCH Threshold SuperSpeed USB VC0 Completion		A2Ch

Register 15-103. A00h, A0Ch, A18h, A24h INCH Threshold Port *x* and SuperSpeed USB VC0 Posted (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Posted credits are used for VC0 Memory Write and Message transactions. Port *x* is Ports 0, 1, and 2. These Ports and the SuperSpeed USB are associated with register offsets A00h, A0Ch, A18h, and A24h, respectively.				
2:0	***Reserved***	RsvdP	No	000b
7:3	**Posted Payload Credit** Default advertised Posted Payload credit. Bit resolution is in units of 8. Each increment provides 8 Posted Payload credits (*for example*, Ah = 80 Posted Payload credits). Each credit means that 16 bytes of storage are reserved for Posted TLP Payload data. Port 0 = 32 Payload Credits Downstream Port = 128 Payload Credits	RWS	Yes	Refer to Description
13:8	**Posted Header Credit** Default advertised Posted Header credit. Bit resolution is 1 for 1. Each increment provides 1 Posted Header credit (*for example*, Ah = 10 Posted Header credits). Each credit means that storage is reserved for the entire Header of a Posted TLP. Port 0 = 16 Header Credits Downstream Port = 22 Header Credits	RWS	Yes	Refer to Description
31:14	***Reserved***	RsvdP	No	0-0h

Register 15-104. A04h, A10h, A1Ch, A28h INCH Threshold Port *x* and SuperSpeed USB VC0 Non-Posted (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Non-Posted credits are used for VC0 Memory Read, I/O Read, I/O Write, Configuration Read, and Configuration Write transactions. Port *x* is Ports 0, 1, and 2. These Ports and the SuperSpeed USB are associated with register offsets A04h, A10h, A1Ch, and A28h, respectively.				
7:0	**Non-Posted Payload Credit** The Non-Posted Payload is stored with the Non-Posted Header; therefore Non-Posted Payload credit is always available. Because of this, the USB 3382 hardwires this field to 00h (infinite credits).	RsvdP	Yes	00h
13:8	**Non-Posted Header Credit** Default advertised Non-Posted Header credit. Bit resolution is 1 for 1. Each increment provides 1 Non-Posted Header credit (*for example*, Ah = 10 Non-Posted Header credits). Each credit means that storage is reserved for the entire Header of a Non-Posted TLP. Port 0 = 10 Header Credits Downstream Port = 16 Header Credits	RWS	Yes	Refer to Description
23:14	***Reserved***	RsvdP	No	0-0h
25:24	***Factory Test Only***	RWS	Yes	00b
27:26	***Reserved***	RsvdP	No	00b
31:28	***Factory Test Only***	RW	Yes	0h

Register 15-105. A08h, A14h, A20h, A2Ch INCH Threshold Port *x* and SuperSpeed USB VC0 Completion (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Completion credits are used for VC0 Memory Read, I/O Read, I/O Write, Configuration Read, and Configuration Write transaction Completions. Port *x* is Ports 0, 1, and 2. These Ports and the SuperSpeed USB are associated with register offsets A08h, A14h, A20h, and A2Ch, respectively.				
2:0	***Reserved***	RsvdP	No	000b
7:3	**Completion Payload Credit** Default advertised Completion Payload credit. Bit resolution is in units of 8. Each increment provides 8 Completion Payload Credits (*for example*, Ah = 80 Completion Payload credits). Each Credit means that 16 bytes of storage are reserved for Completion TLP Payload data. Port 0 = 224 Payload Credits Downstream Port = 128 Payload Credits	RWS	Yes	Refer to Description
13:8	**Completion Header Credit** Default advertised Completion Header credit. Bit resolution is 1 for 1. Each increment provides 1 Completion Header credit (*for example*, Ah = 10 Completion Header credits). Each credit means that storage is reserved for the entire Header of a Completion TLP. Port 0 = 28 Header Credits Downstream Port = 16 Header Credits	RWS	Yes	Refer to Description
31:14	***Reserved***	RsvdP	No	0-0h

15.15.6 Device-Specific Registers – Physical Layer (Offsets B80h – B88h)

This section details the Device-Specific Physical Layer (PHY) **Advertised N_FTS** register, located at offset B84h; the remaining register within this structure are ***Factory Test Only***. Table 15-27 defines the register map.

Other Device-Specific PHY registers are detailed in Section 15.14.2, "Device-Specific Registers – Physical Layer (Offsets 200h – 25Ch)."

Table 15-27. Device-Specific PHY Register Map (Offsets B80h – B88h) (Port 0)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	Offset
Factory Test Only		B80h
Reserved	Advertised N_FTS	B84h
SerDes Test		B88h

Register 15-106. B84h Advertised N_FTS (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
7:0	**Advertised N_FTS** Advertised Number of Fast Training Sets (N_FTS) value to transmit for all Ports and the SuperSpeed USB (in Training Sets). Used, along with Link speed, for determining the L1 Exit Latency (**Link Capability** register *L1 Exit Latency* field (All Ports, offset 74h[14:12])).	RWS	Yes	40h
31:8	***Reserved***	RsvdP	No	0000_00h

Register 15-107. B88h SerDes Test (Port 0)

Bit(s)	Description	Type	Serial EEPROM and I²C	Default
0	**Bit Error Checker**	RW	Yes	0
1	***Factory Test Only***	RW	Yes	0
2	**Data Bus Width Select**	RW	Yes	0
3	**Packet Length Select**	RW	Yes	0
4	***Factory Test Only***	RW	Yes	0
5	**Comma Detection Enable**	RW	Yes	1
6	**Comma Pattern Generator Enable**	RW	Yes	0
7	**TSEQ Pattern G Enable**	RW	Yes	0
9:8	**PRBS Number Select**	RW	Yes	00b
15:10	***Factory Test Only***	RW	Yes	0-0h
16	**SerDes 0 BIST Generator/Checker Enable** SerDes 0 PRBS Enable. Bits SMB_RX_BISTEN_U and SMB_TX_BISTEN_U of L0_PREG_AD20_IN. 0 = Disables PRBS sequence generation/checking on SerDes 0 1 = Enables PRBS sequence generation/checking on SerDes 0 ***Notes:* *This bit and the Physical Layer Test register SerDes 0 User Test Pattern Enable bit (Port 0, offset 228h[28]) are mutually exclusive functions and must not be enabled together for SerDes 0. The logical result of both bits ANDed with one another must be 0.*** ***PRBS transmission should be enabled only when operating as a Loopback Master, or when the LTSSM has returned to the Detect.Quiet substate and the Port and SuperSpeed USB Control register Port 0 Quiet bit (Port 0, offset 234h[20]) is Set.***	RW	Yes	0
17	**SerDes 1 BIST Generator/Checker Enable** SerDes 1 PRBS Enable. Bits SMB_RX_BISTEN_U and SMB_TX_BISTEN_U of L1_PREG_AD20_IN. 0 = Disables PRBS sequence generation/checking on SerDes 1 1 = Enables PRBS sequence generation/checking on SerDes 1 ***Notes:* *This bit and the Physical Layer Test register SerDes 1 User Test Pattern Enable bit (Port 0, offset 228h[29]) are mutually exclusive functions and must not be enabled together for SerDes 1. The logical result of both bits ANDed with one another must be 0.*** ***PRBS transmission should be enabled only when operating as a Loopback Master, or when the LTSSM has returned to the Detect.Quiet substate and the Port's Port and SuperSpeed USB Control register Port x Quiet bit (Port 0, offset 234h[21:20]) is Set.***	RW	Yes	0

Register 15-107. B88h SerDes Test (Port 0) ***(Cont.)***

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
18	**SerDes 2 BIST Generator/Checker Enable** SerDes 2 PRBS Enable. Bits SMB_RX_BISTEN_U and SMB_TX_BISTEN_U of L2_PREG_AD20_IN. 0 = Disables PRBS sequence generation/checking on SerDes 2 1 = Enables PRBS sequence generation/checking on SerDes 2 ***Notes: This bit and the Physical Layer Test register SerDes 2 User Test Pattern Enable bit (Port 0, offset 228h[30]) are mutually exclusive functions and must not be enabled together for SerDes 2. The logical result of both bits ANDed with one another must be 0.*** ***PRBS transmission should be enabled only when operating as a Loopback Master, or when the LTSSM has returned to the Detect.Quiet substate and the Port and SuperSpeed USB Control register Port 2 Quiet bit (Port 0, offset 234h[22]) is Set.***	RW	Yes	0
19	**SuperSpeed USB BIST Generator/Checker Enable** SuperSpeed USB PRBS Enable. Bits SMB_RX_BISTEN_U and SMB_TX_BISTEN_U of L3_PREG_AD20_IN. 0 = Disables PRBS sequence generation/checking on SerDes 2 1 = Enables PRBS sequence generation/checking on SerDes 2 ***Notes: This bit and the Physical Layer Test register SuperSpeed USB User Test Pattern Enable bit (Port 0, offset 228h[31]) are mutually exclusive functions and must not be enabled together for the SuperSpeed USB. The logical result of both bits ANDed with one another must be 0.*** ***PRBS transmission should be enabled only when operating as a Loopback Master, or when the LTSSM has returned to the Detect.Quiet substate and the Port and SuperSpeed USB Control register SuperSpeed USB Quiet bit (Port 0, offset 234h[23]) is Set.***	RW	Yes	0
27:20	***Reserved***	RsvdP	No	00h
31:28	**SerDes RESET_B** Writing 1 to these bits causes the RESET_B and RESET_B_U inputs to the corresponding SerDes to assert for 128 µs. This reset also causes the corresponding LTSSM to return to its initial state. These bits always return a value of 0 when read.	RW1C	Yes	0h

15.16 Advanced Error Reporting Extended Capability Registers (Offsets FB4h – FE8h)

This section details the Advanced Error Reporting Extended Capability registers. Table 15-28 defines the register map.

Note: *The Root Complex Event Collector registers, located at offsets FE0h through FE8h are visible/ available only in Root Complex mode. For further details, refer to Section 7.4.1.3, "Root Complex Event Collector Registers."*

Table 15-28. Advanced Error Reporting Extended Capability Register Map

31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Next Capability Offset (138h)	Capability Version (1h)	PCI Express Extended Capability ID (0001h)	FB4h
Uncorrectable Error Status			FB8h
Uncorrectable Error Mask			FBCh
Uncorrectable Error Severity			FC0h
Reserved		Correctable Error Status	FC4h
Reserved		Correctable Error Mask	FC8h
Advanced Error Capabilities and Control			FCCh
Header Log 0			FD0h
Header Log 1			FD4h
Header Log 2			FD8h
Header Log 3			FDCh
Reserved (Adapter Mode) Root Error Command (Root Complex Mode)			FE0h
Reserved (Adapter Mode) Root Error Status (Root Complex Mode)			FE4h
Reserved (Adapter Mode) Error Source ID (Root Complex Mode)			FE8h

Register 15-108. FB4h Advanced Error Reporting Extended Capability Header (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
15:0	**PCI Express Extended Capability ID**	RO	Yes	0001h
19:16	**Capability Version**	RO	Yes	1h
31:20	**Next Capability Offset** Program to 138h, which addresses the **Power Budget Extended Capability** structure.	RO	Yes	138h

Register 15-109. FB8h Uncorrectable Error Status (All Ports and USB Controller)

Bit(s)	Description	Ports	Type	Serial EEPROM and I^2C	Default
	Note: *If an individual error is masked (corresponding bit in the* ***Uncorrectable Error Mask*** *register (offset FBCh) is Set) when the error is detected, its Error Status bit is still updated; however, an error reporting Message (ERR_FATAL or ERR_NONFATAL) is* ***not*** *sent to the Root Complex, and the* ***Advanced Error Capabilities and Control*** *register First Error Pointer field (offset FCCh[4:0]) and* ***Header Log x*** *registers (offsets FD0h through FDCh) remain unchanged.*				
3:0	***Reserved***		RsvdP	No	0h
4	**Data Link Protocol Error Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
5	**Surprise Down Error Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
11:6	***Reserved***		RsvdP	No	0-0h
12	**Poisoned TLP Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
13	**Flow Control Protocol Error Status** ***Reserved/Not supported***		RO	No	1
14	**Completion Timeout Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
15	**Completer Abort Status**		RW1CS	Yes	0
16	**Unexpected Completion Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
17	**Receiver Overflow Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
18	**Malformed TLP Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
19	**ECRC Error Status** ***Not supported***		RsvdP	No	0
20	**Unsupported Request Error Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
21	***Reserved***		RsvdP	No	0
22	**Uncorrectable Internal Error Status** 0 = No error is detected 1 = Error is detected	0	RW1CS	Yes	0
	Reserved	Otherwise	RsvdP	No	0
31:23	***Reserved***		RsvdP	No	0-0h

Register 15-110. FBCh Uncorrectable Error Mask (All Ports and USB Controller)

Bit(s)	Description	Ports	Type	Serial EEPROM and I^2C	Default
Note: *The bits in this register can be used to mask their respective* ***Uncorrectable Error Status*** *register bits (All Ports and USB Controller, offset FB8h).*					
3:0	***Reserved***		RsvdP	No	0h
4	**Data Link Protocol Error Mask** 0 = No mask is Set 1 = Masks error reporting, first error update, and Header logging for this error		RWS	Yes	0
5	**Surprise Down Error Mask** 0 = No mask is Set 1 = Masks error reporting, first error update, and Header logging for this error		RWS	Yes	0
11:6	***Reserved***		RsvdP	No	0-0h
12	**Poisoned TLP Mask** 0 = No mask is Set 1 = Masks error reporting, first error update, and Header logging for this error		RWS	Yes	0
13	**Flow Control Protocol Error Mask** ***Reserved/Not supported***		RsvdP	No	0
14	**Completion Timeout Mask** 0 = No mask is Set 1 = Masks error reporting, first error update, and Header logging for this error		RWS	Yes	0
15	**Completer Abort Mask**		RWS	Yes	0
16	**Unexpected Completion Mask** 0 = No mask is Set 1 = Masks error reporting, first error update, and Header logging for this error		RWS	Yes	0
17	**Receiver Overflow Mask** 0 = No mask is Set 1 = Masks error reporting, first error update, and Header logging for this error		RWS	Yes	0
18	**Malformed TLP Mask** 0 = No mask is Set 1 = Masks error reporting, first error update, and Header logging for this error		RWS	Yes	0
19	**ECRC Error Mask** ***Not supported***		RsvdP	No	0
20	**Unsupported Request Error Mask** 0 = No mask is Set 1 = Masks error reporting, first error update, and Header logging for this error		RWS	Yes	0
21	***Reserved***		RsvdP	No	0
22	**Uncorrectable Internal Error Mask** 0 = No mask is Set 1 = Masks error reporting, first error update, and Header logging for this error	0	RWS	Yes	1
	Reserved	Otherwise	RsvdP	No	1
31:23	***Reserved***		RsvdP	No	0-0h

Register 15-111. FC0h Uncorrectable Error Severity (All Ports and USB Controller)

Bit(s)	Description	Ports	Type	Serial EEPROM and I²C	Default
3:0	***Reserved***		RsvdP	No	0h
4	**Data Link Protocol Error Severity** 0 = Error is reported as non-fatal 1 = Error is reported as fatal		RWS	Yes	1
5	**Surprise Down Error Severity** 0 = Error is reported as non-fatal 1 = Error is reported as fatal		RWS	Yes	1
11:6	***Reserved***		RsvdP	No	0-0h
12	**Poisoned TLP Severity** 0 = Error is handled as an Advisory Non-Fatal error, and reported as a Correctable error 1 = Error is reported as fatal		RWS	Yes	0
13	**Flow Control Protocol Error Severity**		RO	No	1
14	**Completion Timeout Severity** 0 = Error is handled as an Advisory Non-Fatal error, and reported as a Correctable error 1 = Error is reported as fatal		RWS	Yes	0
15	**Completer Abort Severity** 0 = Error is handled as an Advisory Non-Fatal error, and reported as a Correctable error 1 = Error reported as fatal		RWS	Yes	0
16	**Unexpected Completion Severity** 0 = Error is handled as an Advisory Non-Fatal error, and reported as a Correctable error 1 = Error is reported as fatal		RWS	Yes	0
17	**Receiver Overflow Severity** 0 = Error is reported as non-fatal 1 = Error is reported as fatal		RWS	Yes	1
18	**Malformed TLP Severity** 0 = Error is reported as non-fatal 1 = Error is reported as fatal		RWS	Yes	1
19	**ECRC Error Severity** ***Not supported***		RsvdP	No	0

Register 15-111. FC0h Uncorrectable Error Severity (All Ports and USB Controller) *(Cont.)*

Bit(s)	Description	Ports	Type	Serial EEPROM and I^2C	Default
20	**Unsupported Request Error Severity** 0 = Error is handled as an Advisory Non-Fatal error, and reported as a Correctable error 1 = Error is reported as fatal		RWS	Yes	0
21	***Reserved***		RsvdP	No	0
22	**Uncorrectable Internal Error Severity** 0 = Error is reported as non-fatal 1 = Error is reported as fatal	0	RWS	Yes	1
	Reserved	Otherwise	RsvdP	No	1
31:23	***Reserved***		RsvdP	No	0-0h

Register 15-112. FC4h Correctable Error Status (All Ports and USB Controller)

Bit(s)	Description	Ports	Type	Serial EEPROM and I^2C	Default
Note: *The bits in this register can be masked by their respective* ***Correctable Error Mask*** *register bits (All Ports and USB Controller, offset FC8h).*					
0	**Receiver Error Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
5:1	***Reserved***		RsvdP	No	0-0h
6	**Bad TLP Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
7	**Bad DLLP Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
8	**REPLAY NUM Rollover Status** Replay Number Rollover status. 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
11:9	***Reserved***		RsvdP	No	000b
12	**Replay Timer Timeout Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
13	**Advisory Non-Fatal Error Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
14	**Corrected Internal Error Status** 0 = No error is detected 1 = Error is detected	0	RW1CS	Yes	0
	Reserved	Otherwise	RsvdP	No	0
15	**Header Log Overflow Status** 0 = No error is detected 1 = Error is detected		RW1CS	Yes	0
31:16	***Reserved***		RsvdP	No	0000h

Register 15-113. FC8h Correctable Error Mask (All Ports and USB Controller)

Bit(s)	Description	Ports	Type	Serial EEPROM and I^{2}C	Default
Note: The bits in this register can be used to mask their respective ***Correctable Error Status*** register bits (All Ports and USB Controller, offset *FC4h*).					
0	**Receiver Error Mask** 0 = Error reporting is not masked 1 = Error reporting is masked		RWS	Yes	0
5:1	***Reserved***		RsvdP	No	0-0h
6	**Bad TLP Mask** 0 = Error reporting is not masked 1 = Error reporting is masked		RWS	Yes	0
7	**Bad DLLP Mask** 0 = Error reporting is not masked 1 = Error reporting is masked		RWS	Yes	0
8	**REPLAY NUM Rollover Mask** Replay Number Rollover mask. 0 = Error reporting is not masked 1 = Error reporting is masked		RWS	Yes	0
11:9	***Reserved***		RsvdP	No	000b
12	**Replay Timer Timeout Mask** 0 = Error reporting is not masked 1 = Error reporting is masked		RWS	Yes	0
13	**Advisory Non-Fatal Error Mask** 0 = Error reporting is not masked 1 = Error reporting is masked		RWS	Yes	1
14	**Corrected Internal Error Mask** 0 = Error reporting is not masked 1 = Error reporting is masked	0	RWS	Yes	1
	Reserved	Otherwise	RsvdP	No	0
15	**Header Log Overflow Mask** 0 = Error reporting is not masked 1 = Error reporting is masked		RWS	Yes	1
31:16	***Reserved***		RsvdP	No	0000h

Register 15-114. FCCh Advanced Error Capabilities and Control (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
4:0	**First Error Pointer** Identifies the bit position of the first error reported in the **Uncorrectable Error Status** register (All Ports and USB Controller, offset FB8h).	ROS	No	1Fh
5	**ECRC Generation Capable** ***Not supported***	RsvdP	No	0
6	**ECRC Generation Enable** ***Not supported***	RsvdP	No	0
7	**ECRC Check Capable** ***Not supported***	RsvdP	No	0
8	**ECRC Check Enable** ***Not supported***	RsvdP	No	0
31:9	***Reserved***	RsvdP	No	0-0h

Register 15-115. FD0h Header Log 0 (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
31:0	**TLP Header 0** First DWord Header. TLP Header associated with error.	ROS	Yes	0000_0000h

Register 15-116. FD4h Header Log 1 (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
31:0	**TLP Header 1** Second DWord Header. TLP Header associated with error.	ROS	Yes	0000_0000h

Register 15-117. FD8h Header Log 2 (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
31:0	**TLP Header 2** Third DWord Header. TLP Header associated with error.	ROS	Yes	0000_0000h

Register 15-118. FDCh Header Log 3 (All Ports and USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
31:0	**TLP Header 3** Fourth DWord Header. TLP Header associated with error.	ROS	Yes	0000_0000h

Register 15-119. FE0h Root Error Command (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^{2}C	Default
Note:	*This register is visible/available only in Root Complex mode.* ***Reserved*** *in Adapter mode.*			
0	**Correctable Error Reporting Enable** 0 = Disables 1 = Enables the SuperSpeed USB to report Correctable errors	RW	Yes	0
1	**Non-Fatal Error Reporting Enable** 0 = Disables 1 = Enables the SuperSpeed USB to report Non-Fatal errors	RW	Yes	0
2	**Fatal Error Reporting Enable** 0 = Disables 1 = Enables the SuperSpeed USB to report Fatal errors	RW	Yes	0
31:3	***Reserved***	RsvdP	No	0-0h

Register 15-120. FE4h Root Error Status (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^{2}C	Default
Note:	*This register is visible/available only in Root Complex mode.* ***Reserved*** *in Adapter mode.*			
0	**ERR_COR Received** 0 = SuperSpeed USB did not receive a Correctable error 1 = SuperSpeed USB received a Correctable error	RW1CS	Yes	0
1	**Multiple ERR_COR Received** 0 = SuperSpeed USB did not receive multiple Correctable errors 1 = SuperSpeed USB received multiple Correctable errors	RW1CS	Yes	0
2	**ERR_FATAL/NONFATAL Received** 0 = SuperSpeed USB did not receive a Fatal or Non-Fatal error 1 = SuperSpeed USB received a Fatal or Non-Fatal error	RW1CS	Yes	0
3	**ERR_FATAL/NONFATAL Received** 0 = SuperSpeed USB did not receive a Fatal or Non-Fatal error 1 = SuperSpeed USB received a Fatal or Non-Fatal error	RW1CS	Yes	0
4	**First Uncorrectable Fatal**	RW1CS	Yes	0
5	**Non-Fatal Error Messages Received**	RW1CS	Yes	0
6	**Fatal Error Messages Received**	RW1CS	Yes	0
26:7	***Reserved***	RsvdP	No	0-0h
31:27	**Advanced Error Interrupt Message Number**	RO	Yes	0-0h

Register 15-121. FE8h Error Source ID (USB Controller)

Bit(s)	Description	Type	Serial EEPROM and I^2C	Default
Note: *This register is visible/available only in Root Complex mode.* ***Reserved*** *in Adapter mode.*				
15:0	**ERR_COR Source ID**	ROS	Yes	0000h
31:16	**ERR_FATAL/NONFATAL Source ID**	ROS	Yes	0000h

Chapter 16 USB Configuration Registers

16.1 Introduction

This chapter describes the USB Configuration Space registers (CSRs) that are specific to the USB 3382's USB Controller functions.

These Memory-Mapped CSRs are accessed using the 64-KB Memory space defined by PCI Base Address 0 (**BAR0**; **Base Address 0** register (USB Controller, offset 10h)) in the Type 0 PCI Express Configuration registers. The Indexed registers are accessed by using the **IDXADDR** and **IDXDATA** registers (USB Controller, offsets 30h and 34h, respectively). The USB CSROUT endpoint and the 8051 can also access these CSRs using cursor registers. Each register is 32 bits wide, and is accessed one byte, word, or DWord at a time.

These registers use Little Endian byte ordering, which is consistent with the *PCI Express Base r2.1*. The least significant byte (LSB) in a DWord is accessed at Address 0. The least significant bit (lsb) in a DWord is 0, and the most significant bit (msb) is 31.

After the USB 3382 is powered-up or reset, the registers are programmed to their default values. Writes to unused registers are ignored, and Reads from unused registers return a value of 0. For compatibility with future revisions, ***RsvdZ*** bits within a register must always be written with a 0.

Other registers are defined in Chapter 15, "PCI Configuration Registers."

16.2 Access Attributes

The following table lists the attributes used to indicate the type of access provided by each register bit.

Note: *Register bits that are writeable by the 8051 or USB are also writeable by the Serial EEPROM Controller.*

Table 16-1. Access Attributes

Attribute	Description
HwInit	Hardware or firmware initialized, *such as* by pin strapping or serial EEPROM.
PIN	Read value determined by external pin.
RC	Read-Only. Read to Clear.
RO	Read-Only.
ROS	Sticky Read-Only. Not modified by reset; preserved with AUX power.
RsvdP	***Reserved*** and Preserved. Software must write the Read value.
RsvdZ	***Reserved*** and Zero. Software must write 0.
RW	Read-Write.
RW1C	Write 1 to Clear.
RW1CS	Sticky Write 1 to Clear.
RW1S	Write 1 to Set.
RW1T	Write 1 to toggle.
RWC	Read-Write. Read to Clear.
RWS	Sticky Read-Write.
RWU	**Read/Write (Output Pin), Read-Only (Input Pin)** Specific to the GPIO*x Data* register bits. When the referenced GPIO*x* pin is configured as output, the value written appears on the output pin, and are returned by a Read. When the referenced GPIO*x* pin is configured as an input, Reads to this bit always return the pin state, and Writes have no effect.
RZ	Read value unknown.
RZW1C	Write 1 to Clear; Read value unknown.

16.3 Register Summary

Table 16-2. Register Summary

Register Group	Offset (from BAR0)
USB Controller Device-Specific Registers	00h – 7Ch
USB Interface Control Registers	80h – FFh
PCI Express/Configuration Cursor Registers	100h – 17Fh
DMA Registers	180h – 1FCh 680h – 6BCh
Dedicated Endpoint Registers	200h – 254h
EP 0 and GPEPx Registers	300h – 4FFh
FIFO Registers	500h – 614h
USB Power Management Registers	6C0h – 6C4h
Indexed Registers	Index 00h – FFh

16.4 Preserved Registers during Suspend

The registers listed in Table 16-3 must be powered-up during the Suspend state, and should not be overwritten by the serial EEPROM during a resume re-load.

Table 16-3. Preserved USB Controller Registers during Suspend

Offset	Register	Comment
50h	GPIOCTRL	13 bits; GPIO Control
84h	PRODVENDID	32 bits; Product ID and Vendor ID
88h	RELNUM	16 bits; Device Release Number
8Ch	USBCTL	13 bits; USB Control
90h	USBSTAT	2 bits; USB Status
A4h	OURADDR	7 bits; Our USB Address
A8h	OURCONFIG	8 bits; Our USB Configuration
C8h	USBCTL2	8 bits; USB Control 2
304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h	EP_RSP	9 bits; for all endpoints (Halt and Toggle bits)

16.5 USB Controller Device-Specific Registers

Table 16-4. USB Controller Device-Specific Registers

Offset (from BAR0)	Register	Description
00h	DEVINIT	Device Initialization
04h – 08h	***Reserved***	
0Ch	PCICTL	PCI Control
10h	PCIIRQENB0	PCI Express Interrupt Request Enable 0
14h	PCIIRQENB1	PCI Express Interrupt Request Enable 1
18h	CPUIRQENB0	8051 Interrupt Request Enable 0 (***Reserved*** in Legacy Adapter mode)
1Ch	CPUIRQENB1	8051 Interrupt Request Enable 1 (***Reserved*** in Legacy Adapter mode)
20h	USBIRQENB0	STATIN Interrupt Request Enable 0 (***Reserved*** in Legacy Adapter mode)
24h	USBIRQENB1	STATIN Interrupt Request Enable 1 (***Reserved*** in Legacy Adapter mode)
28h	IRQSTAT0	Interrupt Request Status 0
2Ch	IRQSTAT1	Interrupt Request Status 1
30h	IDXADDR	Indexed Register Address
34h	IDXDATA	Indexed Register Data
38h	FIFOCTL	FIFO Control
3Ch	BAR2CTL	BAR2 Enhanced Control (***Reserved*** in Legacy Adapter mode)
40h	BAR3CTL	BAR3 Enhanced Control (***Reserved*** in Legacy Adapter mode)
44h – 4Ch	***Reserved***	
50h	GPIOCTRL	GPIO Control[a]
54h	GPIOSTAT	GPIO Status[a]
58h	PWMV	GPIO PWM Value[a]
5Ch	PWMRC	GPIO PWM Ramp Control[a]
60h	PWMFREQ	GPIO PWM Clock Frequency[a]
64h – 74h	***Reserved***	
78h	Root Message Dispatch	Root Message Dispatch 0/1 – RCIN/8051 FIFO (***Reserved*** in Adapter mode)
7Ch	HUC Cursor Data	HUC Cursor Data

a. *The GPIO-related registers (USB Controller, offsets 50h through 60h) are* ***reserved*** *(have no function) when the* ***Debug Control*** *register LANE_GOODx#/GPIOx Pin Function Select bit (Port 0, offset 1DCh[22]) is Set.*

Register 16-1. 00h DEVINIT Device Initialization (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**8051 Reset** If a valid serial EEPROM with 8051 firmware is detected, this bit is automatically Cleared when the serial EEPROM Read completes. If a valid serial EEPROM is not detected, this bit is not Cleared. 0 = 8051 is enabled to execute firmware 1 = 8051 is held in a reset state	RWU	Yes	1
1	**USB Soft Reset** Reading this bit always returns a value of 0. 1 = USB Control section of the USB 3382 is reset. Also, the **OURADDR** and **OURCONFIG** registers (USB Controller, offsets A4h and A8h, respectively) are Cleared.	W1RZ	Yes	0
2	**PCI Soft Reset** Reading this bit returns a value of 1 while the Reset sequence is in progress (approximately 10 ms), and a value of 0 when the Reset sequence is complete. 1 = PCI Configuration registers and the PCI Control section of the USB 3382 are reset. Additionally, Hot Reset Training Sets are transmitted from downstream PCI Express Ports.	W1R	Yes	0
3	**Configuration Soft Reset** Reading this bit always returns a value of 0. 1 = All PCI Configuration registers for the USB Controller block are reset	W1RZ	Yes	0
4	**FIFO Soft Reset** Reading this bit always returns a value of 0. 1 = All endpoint FIFOs are flushed	W1RZ	Yes	0
5	**PCI Enable** This bit is automatically Set when a valid serial EEPROM is not detected and Adapter mode is selected. 0 = PCI Express accesses to the USB 3382 result in a Target Retry response 1 = USB 3382 normally responds to PCI Express accesses	RWU	Yes	0
6	**PCI ID** 0 = Standard PCI Device ID and Vendor ID are returned to the PCI Express Host when the **PRODVENDID** register (USB Controller, offset 84h) is accessed 1 = Subsystem Vendor ID and Device ID values in the **Subsystem ID and Subsystem Vendor ID** register (USB Controller, offset 2Ch) are returned to the PCI Express Host when the **PRODVENDID** register (USB Controller, offset 84h) is accessed	RW	Yes	0
7	***Reserved***	RsvdP	No	0

Register 16-1. 00h DEVINIT Device Initialization (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^{2}C	Default
11:8	**Local Clock Frequency** ***No function***	RW	Yes	8h
15:12	***Reserved***	RsvdP	No	0h
20:16	**PCI Expansion ROM Range** Determines the PCI Expansion ROM Base Address register range, in increments of 2 KB. The default corresponds to a 2-KB range, and the maximum range is 64 KB. Bit 16 of this register corresponds to Address bit 11, and bit 20 corresponds to Address bit 15. Starting with bit 16 of the register, as each successive bit is Cleared, the range doubles. The PCI Expansion ROM Base Address register must be a multiple of the range.	RW	Yes	1Fh
31:21	***Reserved***	RO	No	0-0h

Register 16-2. 0Ch PCICTL PCI Control (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^{2}C	Default
0	**PCIBAR0 Enable** PCI Express-to-Configuration Register Address space (**PCIBAR0**) enable. 0 = **PCIBAR0** is disabled 1 = **PCIBAR0** is enabled	RW	Yes	1
1	**PCIBAR1 Enable** PCI Express-to-8051 Memory Address space (**PCIBAR1**) enable. 0 = **PCIBAR1** is disabled 1 = **PCIBAR1** is enabled	RW	Yes	1
31:2	***Reserved***	RsvdP	No	0-0h

Register 16-3. 10h PCIIRQENB0 PCI Express Interrupt Request Enable 0 (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**Endpoint 0 PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on EP 0.	RW	Yes	0
1	**Legacy Adapter Mode** **GPEP0 OUT/IN PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP0	RW	Yes	0
	Enhanced Adapter Mode **GPEP0 OUT PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP0 OUT	RW	Yes	0
2	**Legacy Adapter Mode** **GPEP1 OUT/IN PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP1	RW	Yes	0
	Enhanced Adapter Mode **GPEP1 OUT PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP1 OUT	RW	Yes	0
3	**Legacy Adapter Mode** **GPEP2 OUT/IN PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP2	RW	Yes	0
	Enhanced Adapter Mode **GPEP2 OUT PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP2 OUT	RW	Yes	0
4	**Legacy Adapter Mode** **GPEP3 OUT/IN PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP3	RW	Yes	0
	Enhanced Adapter Mode **GPEP3 OUT PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP3 OUT	RW	Yes	0
6:5	***Reserved***	RW	Yes	00b
7	**Setup Packet PCI Express Interrupt Enable** 1 = Enables ability to generate a PCI Express interrupt when a Setup packet is received from the Host	RW	Yes	0

Register 16-3. 10h PCIIRQENB0 PCI Express Interrupt Request Enable 0 (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
8	**USB to PCI Express TLP Drained on Port 0 PCI Express Interrupt Enable**	RW	Yes	0
9	**USB to PCI Express TLP Drained on Port 1 PCI Express Interrupt Enable**	RW	Yes	0
10	**USB Configuration Retry PCI Express Interrupt Enable**	RW	Yes	0
11	**USB IN FIFO Timeout PCI Express Interrupt Enable**	RW	Yes	0
16:12	***Reserved***	RsvdP	No	0-0h
17	**Legacy Adapter Mode** ***Reserved***	RsvdP	No	0
	Enhanced Adapter Mode **GPEP0 IN PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP0 IN	RW	Yes	0
18	**Legacy Adapter Mode** ***Reserved***	RsvdP	No	0
	Enhanced Adapter Mode **GPEP1 IN PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP1 IN	RW	Yes	0
19	**Legacy Adapter Mode** ***Reserved***	RsvdP	No	0
	Enhanced Adapter Mode **GPEP2 IN PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP2 IN	RW	Yes	0
20	**Legacy Adapter Mode** ***Reserved***	RsvdP	No	0
	Enhanced Adapter Mode **GPEP3 IN PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active on GPEP3 IN	RW	Yes	0
31:21	***Reserved***	RsvdP	No	0-0h

Register 16-4. 14h PCIIRQENB1 PCI Express Interrupt Request Enable 1 (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**SOF PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when the USB 3382 receives a Start-of-Frame (SOF) packet	RW	Yes	0
1	**Resume PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when the USB 3382 resumes from the Suspended state	RW	Yes	0
2	**Suspend Request Change PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when a change in the Suspend Request Interrupt state is detected	RW	Yes	0
3	**Suspend Request PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when a USB Suspend Request from the Host is detected	RW	Yes	0
4	**Root Port Reset PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when a Root Port Reset (Host Port Reset) is detected	RW	Yes	0
5	***Reserved***	RW	Yes	0
6	**Control Status PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an IN or OUT token indicating Control Status is received	RW	Yes	0
7	**VBUS PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when a change is detected on the USB_VBUS input	RW	Yes	0

Register 16-4. 14h PCIIRQENB1 PCI Express Interrupt Request Enable 1 (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
8	**EEPROM Done PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when a Serial EEPROM Read or Write transaction completes	RW	Yes	0
9	**DMA Channel 0 PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active from DMA Channel A	RW	Yes	0
10	**DMA Channel 1 PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active from DMA Channel B	RW	Yes	0
11	**DMA Channel 2 PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active from DMA Channel C	RW	Yes	0
12	**DMA Channel 3 PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when an interrupt is active from DMA Channel D	RW	Yes	0
13	**GPIO PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface to generate when an interrupt is active from one of the GPIO*x* pins	RW	Yes	0
14	**SOF Downcount PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when the SOF Frame Downcount Counter reflects a value of 0 and an SOF is detected	RW	Yes	0
15	***Reserved***	RW	Yes	0
16	**PCI Master Cycle Done PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when a USB- or 8051-initiated PCI Express access completes. *For example*, in the case of a PCI Express Memory Read Request, this interrupt indicates that the Read Request has completed and data is available in **PCIMSTDATA**.	RW	Yes	0
17	***Reserved***	RW	Yes	0
18	**PCI Express Hot Plug PCI Express Interrupt Enable**	RW	Yes	0
19	**PCI Target Abort Received PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when the USB 3382 receives a Completion TLP with Completer Abort (CA) status	RW	Yes	0

Register 16-4. 14h PCIIRQENB1 PCI Express Interrupt Request Enable 1 (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
20	**Master Abort Interrupt Enable** 1 = Enables ability to generate an Interrupt Message when the USB 3382 receives a Completion TLP with Unsupported Request (UR) status	RW	Yes	0
24:21	***Reserved***	RsvdP	No	0h
25	**PCI Parity Error PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when the USB 3382 receives a poisoned TLP	RW	Yes	0
26	***Reserved***	RW	Yes	0
27	**Power State Change PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface when the **PCI Power Management Status and Control** register *Power State* field (All Ports and USB Controller, offset 44h[1:0]) changes	RW	Yes	0
28	**PCI Express DL_DOWN State Change PCI Express Interrupt Enable**	RW	Yes	0
29	**PCI Express Hot Reset PCI Express Interrupt Enable**	RW	Yes	0
30	**PCI Express Endpoint Power Management PCI Express Interrupt Enable** ***Valid only in Adapter mode.*** PCI Express Adapter (endpoint) enable for Power Management PCI Express interrupts.	RW	Yes	0
31	**Global PCI Express Interrupt Enable** 1 = Enables ability to generate an Interrupt Message to the PCI Express interface	RW	Yes	0

Register 16-5. 18h CPUIRQENB0 8051 Interrupt Request Enable 0 (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
Note:	*This register is used only by firmware running on the 8051.* ***Reserved*** *in Legacy Adapter mode.*			
0	**Endpoint 0 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active on EP 0	RW	Yes	0
1	**GPEP0 OUT 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active on GPEP0 OUT	RW	Yes	0
2	**GPEP1 OUT 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active on GPEP1 OUT	RW	Yes	0
3	**GPEP2 OUT 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active on GPEP2 OUT	RW	Yes	0
4	**GPEP3 OUT 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active on GPEP3 OUT	RW	Yes	0
6:5	***Reserved***	RW	Yes	00b
7	**Setup Packet 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when a Setup packet is received from the Host	RW	Yes	0
8	**USB to PCI Express TLP Drained on Port 0 8051 Interrupt Enable**	RW	Yes	0
9	**USB to PCI Express TLP Drained on Port 1 8051 Interrupt Enable**	RW	Yes	0
10	**USB Configuration Retry 8051 Interrupt Enable**	RW	Yes	0
11	**USB IN FIFO Timeout 8051 Interrupt Enable**	RW	Yes	0
16:12	***Reserved***	RsvdP	No	0-0h
17	**GPEP0 IN 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active on GPEP0 IN	RW	Yes	0
18	**GPEP1 IN 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active on GPEP1 IN	RW	Yes	0
19	**GPEP2 IN 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active on GPEP2 IN	RW	Yes	0
20	**GPEP3 IN 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active on GPEP3 IN	RW	Yes	0
31:21	***Reserved***	RsvdP	No	0-0h

Register 16-6. 1Ch CPUIRQENB1 8051 Interrupt Request Enable 1 (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
Note:	*This register is used only by firmware running on the 8051.* ***Reserved*** *in Legacy Adapter mode.*			
0	**SOF 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when the USB 3382 receives a Start-of-Frame (SOF) packet	RW	Yes	0
1	**Resume 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when the USB 3382 resumes from the Suspended state	RW	Yes	0
2	**Suspend Request Change 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when a change in the Suspend Request Interrupt state is detected	RW	Yes	0
3	**Suspend Request 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when a USB Suspend Request from the Host is detected	RW	Yes	0
4	**Root Port Reset 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when a Root Port Reset (Host Port Reset) is detected	RW	Yes	0
5	***Reserved***	RW	Yes	0
6	**Control Status 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an IN or OUT token indicating Control Status is received	RW	Yes	0
7	**VBUS 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when a change is detected on the USB_VBUS input	RW	Yes	0
8	**Serial EEPROM Done 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when a Serial EEPROM Write or Read transaction completes	RW	Yes	0
9	**DMA Channel 0 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active from DMA Channel A	RW	Yes	0
10	**DMA Channel 1 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active from DMA Channel B	RW	Yes	0
11	**DMA Channel 2 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active from DMA Channel C	RW	Yes	0

Register 16-6. 1Ch CPUIRQENB1 8051 Interrupt Request Enable 1 (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
12	**DMA Channel 3 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active from DMA Channel D	RW	Yes	0
13	**GPIO 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when an interrupt is active from one of the GPIO*x* pins	RW	Yes	0
14	**SOF Downcount 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when the SOF Frame Downcount Counter reflects a value of 0 and an SOF is detected	RW	Yes	0
15	***Reserved***	RW	Yes	0
16	**PCI Master Cycle Done 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when a USB- or 8051-initiated PCI Express TLP completes	RW	Yes	0
17	***Reserved***	RW	Yes	0
18	**PCI Express Hot Plug 8051 Interrupt Enable**	RW	Yes	0
19	**PCI Target Abort Received 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when the USB 3382 receives a Completion with Completer Abort (CA) status	RW	Yes	0
20	**PCI Master Abort Received 8051 Interrupt Enable** 1 = Enables an 8051 interrupt when the USB 3382 receives a Completion with Unsupported Request (UR) status	RW	Yes	0
21	**Enhanced Adapter Mode** ***Reserved***	RsvdP	No	0
	Root Complex Mode **Correctable Error Message Received 8051 Interrupt Enable**	RW	Yes	0
22	**Enhanced Adapter Mode** ***Reserved***	RsvdP	No	0
	Non-Fatal Error Message Received 8051 Interrupt Enable	RW	Yes	0
23	**Enhanced Adapter Mode** ***Reserved***	RsvdP	No	0
	Fatal Error Message Received 8051 Interrupt Enable	RW	Yes	0

Register 16-6. 1Ch CPUIRQENB1 8051 Interrupt Request Enable 1 (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
24	**PCI INTA# 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when the PCI INTA# input asserts and Root Complex mode is selected	RW	Yes	0
25	**PCI Parity Error 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when the USB 3382 receives a poisoned TLP	RW	Yes	0
26	***Reserved***	RW	Yes	0
27	**Power State Change 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt when the **PCI Power Management Status and Control** register *Power State* field (All Ports and USB Controller, offset 44h[1:0]) changes	RW	Yes	0
28	**PCI Express DL_DOWN State Change 8051 Interrupt Enable**	RW	Yes	0
29	**PCI Express Hot Reset 8051 Interrupt Enable**	RW	Yes	0
30	**PCI Express Endpoint Power Management 8051 Interrupt Enable** ***Valid only in Enhanced Adapter mode.*** 1 = Enables ability to generate an 8051 interrupt to the PCI Express Adapter (endpoint)	RW	Yes	0
31	**Global 8051 Interrupt Enable** 1 = Enables ability to generate an 8051 interrupt	RW	Yes	0

Register 16-7. 20h USBIRQENB0 STATIN Interrupt Request Enable 0 (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
Note:	*This register is used only by firmware running on the 8051.* ***Valid only in Root Complex mode. Reserved*** *in Adapter mode.*			
0	**Endpoint 0 USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an interrupt is active on USB device EP 0	RW	Yes	0
1	**GPEP0 OUT USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an interrupt is active on GPEP0 OUT	RW	Yes	0
2	**GPEP1 OUT USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint when an interrupt is active on GPEP1 OUT	RW	Yes	0
3	**GPEP2 OUT USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint when an interrupt is active on GPEP2 OUT	RW	Yes	0
4	**GPEP3 OUT USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint when an interrupt is active on GPEP3 OUT	RW	Yes	0
6:5	***Reserved***	RW	Yes	00b
7	**Setup Packet USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when a Setup packet is received from the Host	RW	Yes	0
8	**USB to PCI Express TLP Drained on Port 0 USB Interrupt Enable**	RW	Yes	0
9	**USB to PCI Express TLP Drained on Port 1 USB Interrupt Enable**	RW	Yes	0
10	**USB Configuration Retry USB Interrupt Enable**	RW	Yes	0
11	**USB IN FIFO Timeout USB Interrupt Enable**	RW	Yes	0
16:12	***Reserved***	RsvdP	No	0-0h
17	**GPEP0 IN 8051 Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an interrupt is active on GPEP0 IN	RW	Yes	0
18	**GPEP1 IN 8051 Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an interrupt is active on GPEP1 IN	RW	Yes	0
19	**GPEP2 IN 8051 Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an interrupt is active on GPEP2 IN	RW	Yes	0
20	**GPEP3 IN 8051 Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an interrupt is active on GPEP3 IN	RW	Yes	0
31:21	***Reserved***	RsvdP	No	0-0h

Register 16-8. 24h USBIRQENB1 STATIN Interrupt Request Enable 1 (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
Note:	*This register is used only by firmware running on the 8051.* ***Valid only in Root Complex mode. Reserved*** *in Adapter mode.*			
0	**SOF USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when the USB 3382 receives a Start-of-Frame (SOF) packet	RW	Yes	0
1	**Resume USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when the USB 3382 resumes from the Suspended state	RW	Yes	0
2	**Suspend Request Change USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when a change in the Suspend Request state is detected	RW	Yes	0
3	**Suspend Request USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when a USB Suspend Request from the Host is detected	RW	Yes	0
4	**Root Port Reset USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when a Root Port Reset (Host Port Reset) is detected	RW	Yes	0
5	***Reserved***	RW	Yes	0
6	**Control Status USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an IN or OUT token indicating Control Status is received	RW	Yes	0
7	**VBUS USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when a change is detected on the USB_VBUS input	RW	Yes	0
8	**EEPROM Done USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when a Serial EEPROM Read or Write transaction completes	RW	Yes	0
9	**DMA Channel 0 USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an interrupt is active from this DMA channel	RW	Yes	0
10	**DMA Channel 1 USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an interrupt is active from this DMA channel	RW	Yes	0
11	**DMA Channel 2 USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an interrupt is active from this DMA channel	RW	Yes	0

Register 16-8. 24h USBIRQENB1 STATIN Interrupt Request Enable 1 (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
12	**DMA Channel 3 USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an interrupt is active from this DMA channel	RW	Yes	0
13	**GPIO USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when an interrupt is active from one of the GPIO*x* pins	RW	Yes	0
14	**SOF Downcount USB Interrupt Enable** 1 = Enables a STATIN Endpoint interrupt when the SOF Frame Downcount Counter reflects a value of 0 and an SOF is detected	RW	Yes	0
15	***Reserved***	RW	Yes	0
16	**PCI Master Cycle Done USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when a USB- or 8051-initiated PCI Express TLP completes	RW	Yes	0
17	***Reserved***	RW	Yes	0
18	**PCI Express Hot Plug USB Interrupt Enable**	RW	Yes	0
19	**PCI Target Abort Received USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when the USB 3382 receives a Completion with Completer Abort (CA) status	RW	Yes	0
20	**PCI Master Abort Received USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when the USB 3382 receives a Completion with Unsupported Request (UR) status	RW	Yes	0
21	**Correctable Error Message Received USB Interrupt Enable**	RW	Yes	0
22	**Non-Fatal Error Message Received USB Interrupt Enable**	RW	Yes	0
23	**Fatal Error Message Received USB Interrupt Enable**	RW	Yes	0

Register 16-8. 24h USBIRQENB1 STATIN Interrupt Request Enable 1 (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
24	**PCI INTA# USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when the PCI INTA# input asserts and Root Complex mode is selected	RW	Yes	0
25	**PCI Parity Error USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when the USB 3382 receives a poisoned TLP	RW	Yes	0
26	**PCI INTA# Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when the PCI INTA# input asserts and Root Complex mode is selected	RW	Yes	0
27	**Power State Change USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt when the **PCI Power Management Status and Control** register *Power State* field (All Ports and USB Controller, offset 44h[1:0]) changes	RW	Yes	0
28	**PCI Express DL_DOWN State Change USB Interrupt Enable**	RW	Yes	0
29	**PCI Express Hot Reset USB Interrupt Enable**	RW	Yes	0
30	**PCI Express Endpoint Power Management USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt to the PCI Express Adapter (endpoint)	RW	Yes	0
31	**Global USB Interrupt Enable** 1 = Enables ability to generate a STATIN Endpoint interrupt	RW	Yes	0

Register 16-9. 28h IRQSTAT0 Interrupt Request Status 0 (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**Endpoint 0 Interrupt Status** Conveys the EP 0 interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 30Ch), to determine the cause of the interrupt	RO	No	0
1	**Legacy Adapter Mode** **GPEP0 OUT/IN Interrupt Status** Conveys the GPEP0 interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 32Ch), to determine the cause of the interrupt	RO	No	0
	Enhanced Adapter Mode **GPEP0 OUT Interrupt Status** **GPEP0 OUT Interrupt Status** Conveys the GPEP0 OUT interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 32Ch), to determine the cause of the interrupt	RO	No	0
2	**Legacy Adapter Mode** **GPEP1 OUT/IN Interrupt Status** Conveys the GPEP1 interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 34Ch), to determine the cause of the interrupt	RO	No	0
	Enhanced Adapter Mode **GPEP1 OUT Interrupt Status** Conveys the GPEP1 OUT interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 34Ch), to determine the cause of the interrupt	RO	No	0
3	**Legacy Adapter Mode** **GPEP2 OUT/IN Interrupt Status** Conveys the GPEP2 interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 36Ch), to determine the cause of the interrupt	RO	No	0
	Enhanced Adapter Mode **GPEP2 OUT Interrupt Status** Conveys the GPEP2 OUT interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 36Ch), to determine the cause of the interrupt	RO	No	0

Register 16-9. 28h IRQSTAT0 Interrupt Request Status 0 (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
4	**Legacy Adapter Mode** **GPEP3 OUT/IN Interrupt Status** Conveys the GPEP3 interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 38Ch), to determine the cause of the interrupt	RO	No	0
	Enhanced Adapter Mode **GPEP3 OUT Interrupt Status** Conveys the GPEP3 OUT interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 38Ch), to determine the cause of the interrupt	RO	No	0
6:5	***Reserved***	RO	No	00b
7	**Setup Packet Interrupt Status** Writing 1 Clears this bit. 1 = Setup packet was received from the Host	RW1C	Yes	0
8	**USB to PCI Express TLP Drained on Port 0** 1 = PCI Express TLP generated from USB (PCIOUT) targeting PCI Express Port 0 has been transmitted	RW1C	Yes	0
9	**USB to PCI Express TLP Drained on Port 1** 1 = PCI Express TLP generated from USB (PCIOUT) targeting PCI Express Port 1 has been transmitted	RW1C	Yes	0
10	**USB Configuration Retry Interrupt Status**	RW1C	Yes	0
11	**USB IN FIFO Timeout Interrupt Status**	RW1C	Yes	0

Register 16-9. 28h IRQSTAT0 Interrupt Request Status 0 (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
12	**INTA# Asserted** 1 = INTA# was sent/asserted from the endpoint to the PCI Express interface, through an INT*x* Message or the PEX_INTA# pin	RO	No	0
16:13	***Reserved***	RO	No	0h
17	**Legacy Adapter Mode** ***Reserved***	RsvdP	No	0
	Enhanced Adapter Mode **GPEP0 IN Interrupt Status** Conveys the GPEP0 IN interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 3ECh), to determine the cause of the interrupt	RO	No	0
18	**Legacy Adapter Mode** ***Reserved***	RsvdP	No	0
	Enhanced Adapter Mode **GPEP1 IN Interrupt Status** Conveys the GPEP1 IN interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 40Ch), to determine the cause of the interrupt	RO	No	0
19	**Legacy Adapter Mode** ***Reserved***	RsvdP	No	0
	Enhanced Adapter Mode **GPEP2 IN Interrupt Status** Conveys the GPEP2 IN interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 42Ch), to determine the cause of the interrupt	RO	No	0
20	**Legacy Adapter Mode** ***Reserved***	RsvdP	No	0
	Enhanced Adapter Mode **GPEP3 IN Interrupt Status** Conveys the GPEP3 IN interrupt status. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the endpoint's **EP_STAT** register (USB Controller, offset 44Ch), to determine the cause of the interrupt	RO	No	0
31:21	***Reserved***	RsvdP	No	0-0h

Register 16-10. 2Ch IRQSTAT1 Interrupt Request Status 1 (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**SOF Interrupt Status** Indicates when the USB 3382 receives a Start-of-Frame (SOF) packet. Writing 1 Clears this status bit. This bit is Set every millisecond for full-speed connections, and every 125 µs for high-speed connections.	RW1C	Yes	0
1	**Resume Interrupt Status** Writing 1 Clears this status bit. 1 = Indicates that the USB 3382 resumed from the Suspended state	RW1C	Yes	0
2	**Suspend Request Change Interrupt Status** Writing 1 Clears this status bit. 1 = Suspend Request Interrupt state (bit 3) changed	RW1C	Yes	0
3	**Suspend Request Interrupt Status** The Suspend Request state cannot be Set nor Cleared by writing this bit. Instead, writing 1 to this bit places the USB 3382 into Low-Power Suspend mode. (Refer to Section 14.6, "USB Suspend/Resume – Root Complex Mode," and Section 14.5, "USB Suspend/Resume Sequences – Adapter Mode," for further details.) 1 = USB 3382 detected a USB Suspend Request from the Host	W1R	Yes	0
4	**Root Port Reset Interrupt Status** Writing 1 Clears this status bit. 1 = Indicates a change in the Root Port Reset (Host Port Reset) Detector state	RW1C	Yes	0
5	***Reserved***	RO	No	0
6	**Control Status Interrupt Status** Writing 1 Clears this status bit. 1 = IN or OUT token indicating Control Status was received	RW1C	Yes	0
7	**VBUS Interrupt Status** Writing 1 Clears this status bit. 1 = Indicates that a change occurred on the USB_VBUS input. Read the **USBCTL** register *VBUS Pin* bit (USB Controller, offset 8Ch[10]) for the current state of this input.	RW1C	Yes	0

Register 16-10. 2Ch IRQSTAT1 Interrupt Request Status 1 (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
8	**Serial EEPROM Done Interrupt Status** Writing 1 Clears this status bit. 1 = Serial EEPROM Read or Write transaction completed	RW1C	Yes	0
9	**DMA Channel 0 Interrupt Status** Conveys the interrupt status for DMA Channel A. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the channel's **DMASTAT** register (USB Controller, offset 184h), to determine the cause of the interrupt	RO	No	0
10	**DMA Channel 1 Interrupt Status** Conveys the interrupt status for DMA Channel B. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the channel's **DMASTAT** register (USB Controller, offset 1A4h), to determine the cause of the interrupt	RO	No	0
11	**DMA Channel 2 Interrupt Status** Conveys the interrupt status for DMA Channel C. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the channel's **DMASTAT** register (USB Controller, offset 1C4h), to determine the cause of the interrupt	RO	No	0
12	**DMA Channel 3 Interrupt Status** Conveys the interrupt status for DMA Channel D. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the channel's **DMASTAT** register (USB Controller, offset 1E4h), to determine the cause of the interrupt	RO	No	0
13	**GPIO Interrupt Status** Conveys the interrupt status for the four GPIO*x* pins. This bit is Set independently of the *Interrupt Enable* bit. 1 = Read the **GPIOSTAT** register (USB Controller, offset 54h), to determine the cause of the interrupt	RO	No	0
14	**SOF Downcount Interrupt Status** Writing 1 Clears this status bit. 1 = SOF Frame Downcount Counter reflects a value of 0 and an SOF is detected	RW1C	Yes	0
15	***Reserved***	RO	No	0
16	**PCI Master Cycle Done Interrupt Status** Writing 1 Clears this status bit. 1 = USB- or 8051-initiated PCI Express TLP completed	RW1C	Yes	0
17	***Reserved***	RO	No	0
18	**PCI Express Hot Plug Interrupt Status**	RO	Yes	0
19	**PCI Target Abort Received Interrupt Status** Writing 1 Clears this status bit. 1 = USB 3382 received a Completion with Completer Abort (CA) status	RW1C	Yes	0

Register 16-10. 2Ch IRQSTAT1 Interrupt Request Status 1 (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
20	**PCI Master Abort Received Interrupt Status** Writing 1 Clears this status bit. 1 = USB 3382 received a Completion with Unsupported Request (UR) status	RW1C	Yes	0
21	**Adapter Mode** ***Reserved***	RsvdP	No	0
	Root Complex Mode **Correctable Error Message Received Interrupt Status**	RW1C	Yes	0
22	**Adapter Mode** ***Reserved***	RsvdP	No	0
	Root Complex Mode **Non-Fatal Error Message Received Interrupt Status**	RW1C	Yes	0
23	**Adapter Mode** ***Reserved***	RsvdP	No	0
	Root Complex Mode **Fatal Error Message Received Interrupt Status**	RW1C	Yes	0
24	**PCI INTA# Interrupt Status** Writing 1 Clears this status bit. 1 = PCI INTA# input asserted and Root Complex mode is selected	RW1C	Yes	0
25	**PCI Parity Error Interrupt Status** Writing 1 Clears this status bit. 1 = USB 3382 received a poisoned TLP	RW1C	Yes	0
26	***Reserved***	RO	No	0
27	**Power State Change Interrupt Status** 1 = **PCI Power Management Status and Control** register *Power State* field (All Ports and USB Controller, offset 44h[1:0]) changed	RW1C	Yes	0
28	**PCI Express DL_DOWN State Change Interrupt Status**	RW1C	Yes	0
29	**PCI Express Hot Reset Interrupt Status** 1 = PHY received a Hot Reset on TS1	RW1C	Yes	0
30	**Adapter Mode** **PCI Express Endpoint Power Management Interrupt Status** PCI Express Adapter (endpoint) Power Management PCI interrupt.	RW1C	Yes	0
	Root Complex Mode ***Reserved***	RsvdP	No	0
31	***Reserved***	RO	No	0

Register 16-11. 30h IDXADDR Indexed Register Address (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
15:0	**Indexed Register Address** Selects which indexed register is accessed when the **IDXDATA** register (USB Controller, offset 34h) is read or written.	RW	Yes	0000h
31:16	***Reserved***	RsvdZ	No	0000h

Register 16-12. 34h IDXDATA Indexed Register Data (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
31:0	**Indexed Register Data** Provides access to the Indexed Data register selected by the **IDXADDR** register (USB Controller, offset 30h).	RW	Yes	0000_0000h

Register 16-13. 38h FIFOCTL FIFO Control (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
1:0	**FIFO Configuration Select** ***Reserved***	RW	Yes	00b
2	**Legacy Adapter Mode** **PCI BAR2 Select** 0 = GPEP[3:0] endpoint FIFOs are each assigned one quadrant of **PCI BAR2** space (GPEP0 = 1^{st} quadrant, GPEP1 = 2^{nd} quadrant, GPEP2 = 3^{rd} quadrant, GPEP3 = 4^{th} quadrant) 1 = PCI Writes to and Reads from the **PCI BAR2** space are directed to the GPEP[3:0] endpoint FIFOs, as follows: • Writes to the lower half of the space are directed to the GPEP0 endpoint FIFO • Reads from the lower half of the space are directed to the GPEP1 endpoint FIFO • Writes to the upper half of the space are directed to the GPEP2 endpoint FIFO • Reads from the upper half of the space are directed to the GPEP3 endpoint FIFO	RW	Yes	0
	Enhanced Adapter Mode ***Reserved*** ***Note:*** *The Enhanced Adapter mode equivalent function of this bit is provided by the* ***FIFOCTL*** *register {Endpoint Number, Direction of Endpoint} for Quadrant x of BAR2 bits (USB Controller, offset 3Ch[15:0]).*	RsvdP	No	0
3	**Ignore FIFO Availability** 0 = PCI Express accesses to empty/full FIFOs result in a Retry termination 1 = PCI Express accesses to empty and/or full FIFOs result in a standard termination; however, Read data is undefined, and Write data is ignored	RW	Yes	1
15:4	***Reserved***	RO	Yes	000h
31:16	**Legacy Adapter Mode** **PCI BAR2 Range** Determines the **PCI BAR2** range, in increments of 64 KB. The default corresponds to a 64-KB range. Starting with bit 16, as each successive bit is changed to a value of 0, the range doubles. Value of 0 corresponds to a 4-GB range, thereby causing **PCI BAR2** to be disabled. **PCI BAR2** must be a multiple of the range.	RW	Yes	FFFFh
	Enhanced Adapter Mode ***Reserved*** ***Note:*** *The Enhanced Adapter mode equivalent function of this bit is provided by the* ***FIFOCTL*** *register PCI BAR2 Range field (USB Controller, offset 3Ch[31:16]).*	RsvdP	No	0

Register 16-14. 3Ch BAR2CTL BAR2 Enhanced Control (USB Controller)

<table>
<tr><th>Bit(s)</th><th colspan="2">Description</th><th>Access</th><th>Serial EEPROM and I²C</th><th>Default</th></tr>
<tr><td colspan="6">Note: This register is visible/available only in Enhanced Adapter mode. Reserved in Legacy Adapter mode.</td></tr>
<tr><td rowspan="4">3:0</td><td colspan="2">{Endpoint Number, Direction of Endpoint} for Quadrant 0 of BAR2</td><td rowspan="4">RW</td><td rowspan="4">Yes</td><td rowspan="4">0h</td></tr>
<tr><td>Bit(s)</td><td>Description/Function</td></tr>
<tr><td>0</td><td>Direction
Indicates the IN/OUT direction of the GPEPx endpoint indicated by bits [3:1].</td></tr>
<tr><td>3:1</td><td>Endpoint Number
Indicates the GPEPx number to which Quadrant 0 of PCI BAR2 is mapped.
000b = GPEP0
001b = GPEP1
010b = GPEP2
011b = GPEP3
All other encodings are reserved.</td></tr>
<tr><td rowspan="4">7:4</td><td colspan="2">{Endpoint Number, Direction of Endpoint} for Quadrant 1 of BAR2</td><td rowspan="4">RW</td><td rowspan="4">Yes</td><td rowspan="4">0h</td></tr>
<tr><td>Bit(s)</td><td>Description/Function</td></tr>
<tr><td>4</td><td>Direction
Indicates the IN/OUT direction of the GPEPx endpoint indicated by bits [7:5].</td></tr>
<tr><td>7:5</td><td>Endpoint Number
Indicates the GPEPx number to which Quadrant 1 of PCI BAR2 is mapped.
000b = GPEP0
001b = GPEP1
010b = GPEP2
011b = GPEP3
All other encodings are reserved.</td></tr>
</table>

Register 16-14. 3Ch BAR2CTL BAR2 Enhanced Control (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
11:8	**{Endpoint Number, Direction of Endpoint} for Quadrant 2 of BAR2** **Bit(s) — Description/Function** 8 — **Direction** Indicates the IN/OUT direction of the GPEP*x* endpoint indicated by bits [11:9]. 11:9 — **Endpoint Number** Indicates the GPEP*x* number to which Quadrant 2 of **PCI BAR2** is mapped. 000b = GPEP0 001b = GPEP1 010b = GPEP2 011b = GPEP3 All other encodings are ***reserved***.	RW	Yes	0h
15:12	**{Endpoint Number, Direction of Endpoint} for Quadrant 3 of BAR2** **Bit(s) — Description/Function** 12 — **Direction** Indicates the IN/OUT direction of the GPEP*x* endpoint indicated by bits [15:13]. 15:13 — **Endpoint Number** Indicates the GPEP*x* number to which Quadrant 3 of **PCI BAR2** is mapped. 000b = GPEP0 001b = GPEP1 010b = GPEP2 011b = GPEP3 All other encodings are ***reserved***.	RW	Yes	0h
31:16	**PCI BAR2 Range** Determines the **PCI BAR2** range, in increments of 64 KB. The default corresponds to a 64-KB range. Starting with bit 16, as each successive bit is changed to a value of 0, the range doubles. Value of 0 corresponds to a 4-GB range, thereby causing **PCI BAR2** to be disabled. **PCI BAR2** must be a multiple of the range.	RW	Yes	FFFFh

Register 16-15. 40h BAR3CTL BAR3 Enhanced Control (USB Controller)

<table>
<tr><th>Bit(s)</th><th>Description</th><th>Access</th><th>Serial EEPROM and I^{2}C</th><th>Default</th></tr>
<tr><td colspan="5">Note: This register is visible/available only in Enhanced Adapter mode. Reserved in Legacy Adapter mode.</td></tr>
<tr><td>3:0</td><td>{Endpoint Number, Direction of Endpoint} for Quadrant 0 of BAR3
<table>
<tr><th>Bit(s)</th><th>Description/Function</th></tr>
<tr><td>0</td><td>Direction
Indicates the IN/OUT direction of the GPEPx endpoint indicated by bits [3:1].</td></tr>
<tr><td>3:1</td><td>Endpoint Number
Indicates the GPEPx number to which Quadrant 0 of PCI BAR3 is mapped.
000b = GPEP0
001b = GPEP1
010b = GPEP2
011b = GPEP3
All other encodings are reserved.</td></tr>
</table></td><td>RW</td><td>Yes</td><td>0h</td></tr>
<tr><td>7:4</td><td>{Endpoint Number, Direction of Endpoint} for Quadrant 1 of BAR3
<table>
<tr><th>Bit(s)</th><th>Description/Function</th></tr>
<tr><td>4</td><td>Direction
Indicates the IN/OUT direction of the GPEPx endpoint indicated by bits [7:5].</td></tr>
<tr><td>7:5</td><td>Endpoint Number
Indicates the GPEPx number to which Quadrant 1 of PCI BAR3 is mapped.
000b = GPEP0
001b = GPEP1
010b = GPEP2
011b = GPEP3
All other encodings are reserved.</td></tr>
</table></td><td>RW</td><td>Yes</td><td>0h</td></tr>
</table>

Register 16-15. 40h BAR3CTL BAR3 Enhanced Control (USB Controller) *(Cont.)*

<table>
<tr><th>Bit(s)</th><th>Description</th><th>Access</th><th>Serial EEPROM and I²C</th><th>Default</th></tr>
<tr><td>11:8</td><td>{Endpoint Number, Direction of Endpoint} for Quadrant 2 of BAR3
<table>
<tr><th>Bit(s)</th><th>Description/Function</th></tr>
<tr><td>8</td><td>Direction
Indicates the IN/OUT direction of the GPEPx endpoint indicated by bits [11:9].</td></tr>
<tr><td>11:9</td><td>Endpoint Number
Indicates the GPEPx number to which Quadrant 2 of PCI BAR3 is mapped.
000b = GPEP0
001b = GPEP1
010b = GPEP2
011b = GPEP3
All other encodings are reserved.</td></tr>
</table></td><td>RW</td><td>Yes</td><td>0h</td></tr>
<tr><td>15:12</td><td>{Endpoint Number, Direction of Endpoint} for Quadrant 3 of BAR3
<table>
<tr><th>Bit(s)</th><th>Description/Function</th></tr>
<tr><td>12</td><td>Direction
Indicates the IN/OUT direction of the GPEPx endpoint indicated by bits [15:13].</td></tr>
<tr><td>15:13</td><td>Endpoint Number
Indicates the GPEPx number to which Quadrant 3 of PCI BAR3 is mapped.
000b = GPEP0
001b = GPEP1
010b = GPEP2
011b = GPEP3
All other encodings are reserved.</td></tr>
</table></td><td>RW</td><td>Yes</td><td>0h</td></tr>
<tr><td>31:16</td><td>PCI BAR3 Range
Determines the PCI BAR3 range, in increments of 64 KB. The default corresponds to a 64-KB range. Starting with bit 16, as each successive bit is changed to a value of 0, the range doubles. Value of 0 corresponds to a 4-GB range, thereby causing PCI BAR3 to be disabled. PCI BAR3 must be a multiple of the range.</td><td>RW</td><td>Yes</td><td>FFFFh</td></tr>
</table>

Register 16-16. 50h GPIOCTRL GPIO Control (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I2C	Default
*Notes: This register is **reserved** (has no function) when the **Debug Control** register LANE_GOODx#/GPIOx Pin Function Select bit (Port 0, offset 1DCh[22]) is Set.* *Bits [11:8] can be used to mask their respective **GPIO Status** register GPIOx Interrupt bit (USB Controller, offset 54h[3:0], respectively).*				
0	**GPIO0 Data** When the GPIO0 pin is programmed as an input (bit 4 is Cleared), reading this bit returns the value present on the GPIO0 pin. When the GPIO0 pin is programmed as an output (bit 4 is Set), values written to this bit appear on the GPIO0 pin. Reading this bit returns the previously written value.	RWU	Yes	0
1	**GPIO1 Data** When the GPIO1 pin is programmed as an input (bit 5 is Cleared), reading this bit returns the value present on the GPIO1 pin. When the GPIO1 pin is programmed as an output (bit 5 is Set), values written to this bit appear on the GPIO1 pin. Reading this bit returns the previously written value.	RWU	Yes	0
2	**GPIO2 Data** When the GPIO2 pin is programmed as an input (bit 6 is Cleared), reading this bit returns the value present on the GPIO2 pin. When the GPIO2 pin is programmed as an output (bit 6 is Set), values written to this bit appear on the GPIO2 pin. Reading this bit returns the previously written value.	RWU	Yes	0
3	**GPIO3 Data** When the GPIO3 pin is programmed as an input (bit 7 is Cleared), reading this bit returns the value present on the GPIO3 pin. When the GPIO3 pin is programmed as an output (bit 7 is Set), values written to this bit appear on the GPIO3 pin. Reading this bit returns the previously written value.	RWU	Yes	0
4	**GPIO0 Output Enable** 0 = GPIO0 pin is an input 1 = GPIO0 pin is an output	RWS	Yes	1
5	**GPIO1 Output Enable** 0 = GPIO1 pin is an input 1 = GPIO1 pin is an output	RWS	Yes	1
6	**GPIO2 Output Enable** 0 = GPIO2 pin is an input 1 = GPIO2 pin is an output	RWS	Yes	0
7	**GPIO3 Output Enable** 0 = GPIO3 pin is an input 1 = GPIO3 pin is an output	RWS	Yes	0

Register 16-16. 50h GPIOCTRL GPIO Control (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^{2}C	Default
8	**GPIO0 Interrupt Enable** 0 = Masks the ability for the GPIO0 input to generate an interrupt 1 = When the GPIO0 pin is programmed as an input (bit 4 is Cleared), changes on the GPIO0 pin are enabled to generate an interrupt	RW	Yes	0
9	**GPIO1 Interrupt Enable** 0 = Masks the ability for the GPIO1 input to generate an interrupt 1 = When the GPIO1 pin is programmed as an input (bit 5 is Cleared), changes on the GPIO1 pin are enabled to generate an interrupt	RW	Yes	0
10	**GPIO2 Interrupt Enable** 0 = Masks the ability for the GPIO2 input to generate an interrupt 1 = When the GPIO2 pin is programmed as an input (bit 6 is Cleared), changes on the GPIO2 pin are enabled to generate an interrupt	RW	Yes	0
11	**GPIO3 Interrupt Enable** 0 = Masks the ability for the GPIO3 input to generate an interrupt 1 = When the GPIO3 pin is programmed as an input (bit 7 is Cleared), changes on the GPIO3 pin are enabled to generate an interrupt	RW	Yes	0
12	**GPIO3 LED Select** Selects between GPIO3 and USB_LINK_ACTIVE function (default). 0 = USB_LINK_ACTIVE (default). USB_LINK_ACTIVE asserts High when the USB Link is transferring data. 1 = GPIO3 pin is driven High during USB activity. The GPIO3 pin must be programmed as an output (bit 7 is Set), for this feature to be available.	RW	Yes	1
15:13	***Reserved***	RsvdP	No	000b

Register 16-16. 50h GPIOCTRL GPIO Control (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
16	**GPIO0 Input De-Bounce Enable** 1 = When the GPIO0 pin is programmed as an input (bit 4 is Cleared), turns On the de-bounce circuit at GPIO0. The de-bounce time is 1.0 to 1.3 ms.	RW	Yes	0
17	**GPIO1 Input De-Bounce Enable** 1 = When the GPIO1 pin is programmed as an input (bit 5 is Cleared), turns On the de-bounce circuit at GPIO1. The de-bounce time is 1.0 to 1.3 ms.	RW	Yes	0
18	**GPIO2 Input De-Bounce Enable** 1 = When the GPIO2 pin is programmed as an input (bit 6 is Cleared), turns On the de-bounce circuit at GPIO2. The de-bounce time is 1.0 to 1.3 ms.	RW	Yes	0
19	**GPIO3 Input De-Bounce Enable** 1 = When the GPIO3 pin is programmed as an input (bit 7 is Cleared), turns On the de-bounce circuit at GPIO3. The de-bounce time is 1.0 to 1.3 ms.	RW	Yes	0
23:20	***Reserved***	RsvdP	No	0h
24	**GPIO0 PWM Enable** 1 = When the GPIO0 pin is programmed as an output (bit 4 is Set), turns On the pulse-width-modulated (PWM) output at GPIO0	RW	Yes	0
25	**GPIO1 PWM Enable** 1 = When the GPIO1 pin is programmed as an output (bit 5 is Set), turns On the PWM output at GPIO1	RW	Yes	0
26	**GPIO2 PWM Enable** 1 = When the GPIO2 pin is programmed as an output (bit 6 is Set), turns On the PWM output at GPIO2	RW	Yes	0
27	**GPIO3 PWM Enable** 1 = When the GPIO3 pin is programmed as an output (bit 7 is Set), turns On the PWM output at GPIO3	RW	Yes	0
31:28	***Reserved***	RsvdP	No	0h

Register 16-17. 54h GPIOSTAT GPIO Status (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
Notes: *This register is **reserved** (has no function) when the **Debug Control** register LANE_GOODx#/GPIOx Pin Function Select bit (Port 0, offset 1DCh[22]) is Set.* *Bits [3:0] can be masked by their respective **GPIO Control** register GPIOx Interrupt Enable bit (USB Controller, offset 50h[11:8], respectively).*				
0	**GPIO0 Interrupt** Writing 1 Clears this bit. De-bounce delays Setting of this bit when the **GPIO Control** register *GPIO0 Input De-Bounce Enable* bit (USB Controller, offset 50h[16]) is Set. 1 = GPIO0 pin state changed from input, to output (**GPIO Control** register *GPIO0 Output Enable* bit (USB Controller, offset 50h[4]) is Set, after being programmed as input (Cleared))	RW1C	Yes	PCFG
1	**GPIO1 Interrupt** Writing 1 Clears this bit. De-bounce delays Setting of this bit when the **GPIO Control** register *GPIO1 Input De-Bounce Enable* bit (USB Controller, offset 50h[17]) is Set. 1 = GPIO1 pin state changed from input, to output (**GPIO Control** register *GPIO1 Output Enable* bit (USB Controller, offset 50h[5]) is Set, after being programmed as input (Cleared))	RW1C	Yes	PCFG
2	**GPIO2 Interrupt** Writing 1 Clears this bit. De-bounce delays Setting of this bit when the **GPIO Control** register *GPIO2 Input De-Bounce Enable* bit (USB Controller, offset 50h[18]) is Set. 1 = GPIO2 pin state changed from input, to output (**GPIO Control** register *GPIO2 Output Enable* bit (USB Controller, offset 50h[6]) is Set, after being programmed as input (Cleared))	RW1C	Yes	PCFG
3	**GPIO3 Interrupt** Writing 1 Clears this bit. De-bounce delays Setting of this bit when the **GPIO Control** register *GPIO3 Input De-Bounce Enable* bit (USB Controller, offset 50h[19]) is Set. 1 = GPIO3 pin state changed from input, to output (**GPIO Control** register *GPIO3 Output Enable* bit (USB Controller, offset 50h[7]) is Set, after being programmed as input (Cleared))	RW1C	Yes	PCFG
31:4	***Reserved***	RsvdP	No	0000_000h

Register 16-18. 58h PWMV GPIO PWM Value (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
Values programmed into this register determine the quantity of steps (out of 256) that the output is driven High. For the remaining steps in the PWM cycle, the output is driven Low. The PWM value for each GPIO can be incremented or decremented by one step every *n* PWM cycles, where *n* is the value programmed into the **GPIO PWM Ramp Control** register (USB Controller, offset 5Ch). ***Note:*** *This register is* ***reserved*** *(has no function) when the* ***Debug Control*** *register LANE_GOODx#/GPIOx Pin Function Select bit (Port 0, offset 1DCh[22]) is Set.*				
7:0	**GPIO0 PWM Value** Programming *n* outputs a waveform with *n*/255 cycles High and (255-*n*)/255 steps Low. Writes of 0 to the **GPIO Control** register *GPIO0 Output Enable* bit (USB Controller, offset 50h[4]) program the value of this field to 00h. Writes of 1 to the *GPIO0 Output Enable* bit program the value of this field to FFh.	RWU	Yes	00h
15:8	**GPIO1 PWM Value** Programming *n* outputs a waveform with *n*/255 cycles High and (255-*n*)/255 steps Low. Writes of 0 to the **GPIO Control** register *GPIO1 Output Enable* bit (USB Controller, offset 50h[5]) program the value of this field to 00h. Writes of 1 to the *GPIO1 Output Enable* bit program the value of this field to FFh.	RWU	Yes	00h
23:16	**GPIO2 PWM Value** Programming *n* outputs a waveform with *n*/255 cycles High and (255-*n*)/255 steps Low. Writes of 0 to the **GPIO Control** register *GPIO2 Output Enable* bit (USB Controller, offset 50h[6]) program the value of this field to 00h. Writes of 1 to the *GPIO2 Output Enable* bit program the value of this field to FFh.	RWU	Yes	00h
31:24	**GPIO3 PWM Value** Programming *n* outputs a waveform with *n*/255 cycles High and (255-*n*)/255 steps Low. Writes of 0 to the **GPIO Control** register *GPIO3 Output Enable* bit (USB Controller, offset 50h[7]) program the value of this field to 00h. Writes of 1 to the *GPIO3 Output Enable* bit program the value of this field to FFh.	RWU	Yes	00h

Register 16-19. 5Ch PWMRC GPIO PWM Ramp Control (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
PWM ramping starts when a value of *n*=1 to 255 is programmed, and ends when the PWM value reaches 255 or 0. The *n* value is used as input to the **GPIO PWM Value** register (USB Controller, offset 58h). ***Note:*** *This register is* ***reserved*** *(has no function) when the* ***Debug Control*** *register LANE_GOODx#/GPIOx Pin Function Select bit (Port 0, offset 1DCh[22]) is Set.*				
6:0	**GPIO0 PWM Ramp Period** For every Ramp Period (RP) of complete PWM cycles, the GPIO0 PWM value is increased/decreased by 1. 00h = PWM ramp is disabled	RW	Yes	00h
7	**GPIO0 PWM Ramp Sign Bit** The value of 0/1 controls that the GPIO0 PWM value is incremented/ decremented after every "RP of complete PWM cycles."	RW	Yes	0
14:8	**GPIO1 PWM Ramp Period** For every RP of complete PWM cycles, the GPIO1 PWM value is increased/ decreased by 1. 00h = PWM ramp is disabled	RW	Yes	00h
15	**GPIO1 PWM Ramp Sign Bit** The value of 0/1 controls that the GPIO1 PWM value is incremented/ decremented after every "RP of complete PWM cycles."	RW	Yes	0
22:16	**GPIO2 PWM Ramp Period** For every RP of complete PWM cycles, the GPIO2 PWM value is increased/ decreased by 1. 00h = PWM ramp is disabled	RW	Yes	00h
23	**GPIO2 PWM Ramp Sign Bit** The value of 0/1 controls that the GPIO2 PWM value is incremented/ decremented after every "RP of complete PWM cycles."	RW	Yes	0
30:24	**GPIO3 PWM Ramp Period** For every RP of complete PWM cycles, the GPIO3 PWM value is increased/ decreased by 1. 00h = PWM ramp is disabled	RW	Yes	00h
31	**GPIO3 PWM Ramp Sign Bit** The value of 0/1 controls that the GPIO3 PWM value is incremented/ decremented after every "RP of complete PWM cycles."	RW	Yes	0

Register 16-20. 60h PWMFREQ GPIO PWM Clock Frequency (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
Note: *This register is* ***reserved*** *(has no function) when the* ***Debug Control*** *register LANE_GOODx#/GPIOx Pin Function Select bit (Port 0, offset 1DCh[22]) is Set.*				
7:0	**PWM Clock Divider** Controls the PWM step frequency (duration of each step). PWM functions repeat in cycles of 256 steps. The value in this register divides an input clock of 62.5 MHz by a programmed value of 1 to 256 (where 0 = 256). Because every PWM cycle has 256 steps, the fastest/slowest PWM base frequency is 245.1 KHz/1.908 KHz, respectively. 00h = 1.908 KHz (slowest) 01h = 245.1KHz (fastest)	RW	Yes	00h
31:8	***Reserved***	RsvdP	No	0000_00h

Register 16-21. 78h Root Message Dispatch Root Message Dispatch 0/1 – RCIN/8051 FIFO (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
Note: ***Valid only in Root Complex mode. Reserved*** *in Adapter mode.*				
0	**Correctable Error Message Dispatch**	RW	Yes	0
1	**Non-Fatal Error Message Dispatch**	RW	Yes	0
2	**Fatal Error Message Dispatch**	RW	Yes	0
3	**MSI Dispatch**	RW	Yes	0
4	**INTA# Message Dispatch**	RW	Yes	0
5	**INTB# Message Dispatch**	RW	Yes	0
6	**INTC# Message Dispatch**	RW	Yes	0
7	**INTD# Message Dispatch**	RW	Yes	0
8	**PME Message Dispatch**	RW	Yes	0
31:9	***Reserved***	RsvdP	No	0-0h

Register 16-22. 7Ch HUC Cursor Data HUC Cursor Data (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
31:0	**HUC Cursor Data**	RW	Yes	0000_0000h

16.6 USB Interface Control Registers

Table 16-5. USB Interface Control Registers

Offset (from BAR0)	Register	Description
80h	STDRSP	Standard Response Control
84h	PRODVENDID	Product and Vendor IDs
88h	RELNUM	Device Release Number
8Ch	USBCTL	USB Control
90h	USBSTAT	USB Status
94h	XCVRDIAG	USB Transceiver Diagnostic Control
98h	SETUPDW0	Setup DWord 0
9Ch	SETUPDW1	Setup DWord 1
A0h	***Reserved***	
A4h	OURADDR	Our USB Address
A8h	A8h	Our USB Configuration
ACh – B0h	***Reserved***	
B4h	USB_CLASS	USB Class, Sub-Class, Protocol
B8h	SS_SEL	SuperSpeed System Exit Latency
BCh	SS_DEL	SuperSpeed Device Exit Latency
C0h	USB2LPM	USB2 Link Power Management
C4h	USB3BELT	USB3 Best Effort Latency Tolerance
C8h	USBCTL2	USB Control 2
CCh	IN_TIMEOUT	IN Endpoint Credit Timeout
D0h	ISODELAY	Isochronous Delay
D4h – FFh	***Reserved***	

Register 16-23. 80h STDRSP Standard Response Control (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
0	**Get Device Status** 0 = Get Device Status Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	1
1	**Get Interface Status** 0 = Get Interface Status Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	1
2	**Get Endpoint Status** 0 = Get Endpoint Status Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	1
3	**Get Device Descriptor** 0 = Get Device Descriptor Request is passed to the CPU, through the Setup registers. 1 = Request is automatically handled, without notifying the CPU. Other Configuration registers in the USB 3382 determine the values in the Descriptor.	RW	Yes	1
4	**Get Configuration Descriptor** 0 = Get Configuration Descriptor Request is passed to the CPU, through the Setup registers. 1 = Request is automatically handled, without notifying the CPU. Other Configuration registers in the USB 3382 determine the values in the Descriptor.	RW	Yes	1
5	**Get/Set Configuration** 0 = Get/Set Configuration Requests are passed to the CPU, through the Setup registers 1 = Requests are automatically handled, without notifying the CPU	RW	Yes	1
6	**Get/Set Interface** 0 = Get/Set Interface Requests are passed to the CPU, through the Setup registers 1 = Requests are automatically handled without notifying the CPU	RW	Yes	1
7	***Reserved***	RW	Yes	1
8	**Get String Descriptor 0** 0 = Get String Descriptor 0 (Language ID) Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	1
9	**Get String Descriptor 1** 0 = Get String Descriptor 1 (Manufacturer ID) Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	1
10	**Get String Descriptor 2** 0 = Get String Descriptor 2 (Product ID) Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	1
11	**Device SET/CLR Device Remote Wake-up** 0 = Device SET/CLR Feature Requests for controlling Device Remote Wakeup Enable are passed to the CPU, through the Setup registers 1 = Requests are automatically handled, without notifying the CPU	RW	Yes	1

Register 16-23. 80h STDRSP Standard Response Control (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
12	**Endpoint SET/CLR Halt** 0 = Endpoint SET/CLR feature Requests for controlling the Halt condition are passed to the CPU, through the Setup registers 1 = Requests are automatically handled, without notifying the CPU	RW	Yes	1
13	**Set Address** 0 = Set Address Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	1
14	**Get Device Qualifier** 0 = Get Device Qualifier Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	1
15	**Get Other Speed Configuration** 0 = Get Other Speed Configuration Descriptor Request is passed to the CPU, through the Setup registers 1 = Request is handled automatically without notifying the CPU	RW	Yes	1
16	**Set Test Mode** 0 = Set Test Mode Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	1
17	**SET/CLR Function Suspend** 0 = Interface SET/CLR features Requests for controlling Function Suspend are passed to the CPU, through the Setup registers 1 = Requests are automatically handled, without notifying the CPU	RW	Yes	1
18	**SET/CLR U1 Enable** 0 = Device SET/CLR features Requests for controlling U1_Enable are passed to the CPU, through the Setup registers 1 = Requests are automatically handled, without notifying the CPU	RW	Yes	1
19	**SET/CLR U2 Enable** 0 = Device SET/CLR features Requests for controlling U2_Enable are passed to the CPU, through the Setup registers 1 = Requests are automatically handled, without notifying the CPU	RW	Yes	1

Register 16-23. 80h STDRSP Standard Response Control (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
20	**SET/CLR LTM Enable** 0 = Device SET/CLR features Requests for controlling Latency Tolerance Messages are passed to the CPU, through the Setup registers 1 = Requests are automatically handled, without notifying the CPU	RW	Yes	1
21	**Get BOS Descriptor** 0 = Get Binary Device Object Store (BOS) Descriptor Request is passed to the CPU, through the Setup registers. 1 = Request is automatically handled, without notifying the CPU. Other Configuration registers in the USB 3382 determine the values in the Descriptor.	RW	Yes	1
22	**Set SEL** 0 = Set “System Exit Latency” Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	1
23	**Get String Descriptor 3** 0 = Get String Descriptor 3 (Serial Number) Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	0
24	**Set Isochronous Delay** 0 = Set Isochronous Delay Request is passed to the CPU, through the Setup registers 1 = Request is automatically handled, without notifying the CPU	RW	Yes	1
30:25	***Reserved***	RsvdZ	Yes	0-0h
31	**Stall Unsupported Requests** 0 = Standard, Class, and Vendor Requests from the USB Host that are not included in bits [24:0] of this register are passed to the CPU firmware 1 = Standard, Class, and Vendor Requests from the USB Host that are not included in bits [24:0] of this register cause a STALL handshake, and EP 0 is halted	RW	Yes	1

Register 16-24. 84h PRODVENDID Product and Vendor IDs (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
15:0	**Vendor ID** ***Used only when the Get Device Descriptor Request is in Auto-Enumerate mode.*** Determines the Vendor ID during a Get Device Descriptor Request.	RW	Yes	0525h
31:16	**Product ID** ***Used only when the Get Device Descriptor Request is in Auto-Enumerate mode.*** Determines the Product ID during a Get Device Descriptor Request.	RW	Yes	3382h

Register 16-25. 88h RELNUM Device Release Number (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
15:0	**Device Release Number** ***Used only when the Get Device Descriptor Request is in Auto-Enumerate mode.*** Determines the USB 3382 Release Number during a Get Device Descriptor Request. ***Note:*** *The default value of RELNUM is based upon the Silicon Revision, encoded as a four-digit binary-coded decimal (BCD) value. The value of RELNUM for the first USB 3382 release is 0101h. The two least-significant digits are incremented for mask changes, and the two most-significant digits are incremented for major revisions. This value can be changed by the 8051 to implement an application-specific Release Number. This value can also be modified by serial EEPROM, I²C, and/or the PCI Express interface prior to USB enumeration.*	RW	Yes	0101hSilicon Revision AA: 0101h Silicon Revision AB: 0102h
31:16	***Reserved***	RsvdZ	Yes	0000h

Register 16-26. 8Ch USBCTL USB Control (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
0	**Self-Powered Status** Determines the value of the *Self-Powered* bit in the Device Status Request, when operating in Auto-Enumerate mode.	RW	Yes	1 (Adapter mode) 0 (Root Complex mode)
1	**Remote Wakeup Enable** 1 = Enables the Device Remote Wakeup feature ***USB r2.0*** This bit is controlled by the Device Remote Wakeup SET/CLR Standard Request, and is reported to the USB Host in the *GetStatus Device*, bit 1. ***USB r3.0*** This bit is the Function Remote Wakeup. It is controlled by the *SET/CLR Feature Interface Suspend*, bit 1, and is reported to the USB Host in the *GetStatus Interface*, bit 1.	RW	Yes	0
2	**PCI Express Wakeup Enable** 1 = Enables the PCI Express WAKE# pin or beacon, to wake up the USB 3382	RW	Yes	0
3	**USB Detect Enable** 0 = USB 3382 does not appear to be connected to the USB Host. 1 = USB 3382 appears to be connected to the USB Host. This bit should not be Set until the Configuration registers are programmed. This bit is automatically Set if Root Complex mode is selected, and a valid serial EEPROM is not detected at reset time.	RW	Yes	0
4	**PME Polarity** Not used in PCI Express environments.	RW	Yes	0
5	**Remote Wakeup Support** Indicates whether the USB 3382 supports Device Remote Wakeup. ***USB r2.0*** Reported to the USB Host in the USB Configuration Descriptor's *bmAttributes* field, bit 5, during Auto-Enumerate mode. ***USB r3.0*** Reported to the USB Host in the USB Configuration Descriptor's *bmAttributes* field, bit 5, during Auto-Enumerate mode. Also reported in the *GetStatus Interface*, bit 0.	RW	Yes	0
6	**Self-Powered USB Device** Reported to the USB Host in the USB Configuration Descriptor's *bmAttributes* field, bit 6, during Auto-Enumerate mode, and indicates whether the USB 3382 is self-powered.	RW	Yes	1 (Adapter mode) 0 (Root Complex mode)
7	**Immediately Suspend** 0 = **IRQSTAT1** register *Suspend Request Interrupt Status* bit (USB Controller, offset 2Ch[3]) must be written to initiate the Suspend sequence 1 = Allows the USB 3382 to automatically enter the Suspend state when the USB is idle for 3 ms. Automatically Set if Root Complex mode is selected and a valid serial EEPROM is not detected at reset time.	RW	Yes	0

Register 16-26. 8Ch USBCTL USB Control (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
8	***Reserved***	RW	Yes	0
9	**Timed Disconnect** When written, bit 3 (*USB Detect Enable)* is disabled after a 20-ms delay, effectively disconnecting the USB 3382 from the USB Host. After a 1s delay, bit 3 is Set again, re-connecting the USB 3382 to the USB Host.	RW1S	Yes	0
10	**VBUS Pin** Indicates the USB_VBUS input state. 1 = Indicates that the USB 3382 is connected to the USB Host	PIN	Yes	0
11	**USB Root Port Wakeup Enable** 0 = Wakeup condition is not detected 1 = Root Port Wakeup condition is detected when activity is detected on the USB line interface	RW	Yes	1
12	**Vendor ID String Enable** 0 = *Vendor String Index* value in the Device Descriptor is Cleared, and the string Descriptor Read is acknowledged with a stall in Auto-Enumerate mode 1 = Allows the default Vendor String Descriptor to be returned to the Host in Auto-Enumerate mode	RW	Yes	1
13	**Product ID String Enable** 0 = *Product String Index* value in the Device Descriptor is Cleared, and the string Descriptor Read is acknowledged with a stall in Auto-Enumerate mode 1 = Allows the default Product String Descriptor to be returned to the Host in Auto-Enumerate mode	RW	Yes	1
31:14	***Reserved***	RsvdZ	Yes	0-0h

Register 16-27. 90h USBSTAT USB Status (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
3:0	***Reserved***	RsvdZ	Yes	0h
4	***Reserved***	RO	Yes	0
5	**Generate Resume** Writing 1 initiates a Resume sequence to the Host, if Device Remote Wakeup is enabled (**USBCTL** register *Remote Wakeup Enable* bit (USB Controller, offset 8Ch[1]) is Set). This bit is self-Clearing, and always returns a value of 0.	RW1S	Yes	0
6	**Full-Speed Mode** 1 = USB 3382 is operating in Full-Speed mode (12 Mbps)	RO	Yes	0
7	**High-Speed Mode** 1 = USB 3382 is operating in High-Speed mode (480 Mbps)	RO	Yes	0
8	**SuperSpeed Mode** 1 = USB 3382 is operating in SuperSpeed mode (5 Gbps)	RO	Yes	0
9	**Suspend Status** Automatically Set when the USB 3382 enters the Suspend state. Writing 1 Clears this bit. 1 = Indicates that the USB 3382 was previously in the Suspend state	RW1C	Yes	0
10	**Remote Wakeup Status** Writing 1 Clears this bit. 1 = Indicates that a remote Wakeup beacon or WAKE# input was received from a PCI Express device. Causes a Function Wake Device Notification TP to be sent to the *USB r3.0* Host when resume is finished and the Link has returned to the L0 Link PM state.	RW1C	Yes	0
11	**Enhanced Mode** If programming this register from serial EEPROM, always write 0 to this bit. 0 = USB 3382 is operating in Legacy Adapter mode 1 = USB 3382 is operating in Enhanced Adapter mode	RO	Yes	0
12	**Host Mode** If programming this register from serial EEPROM, always write 0 to this bit. 0 = USB 3382 is operating in Adapter mode 1 = USB 3382 is operating in Root Complex mode (known as Host mode, in the legacy NET 2282)	RO	Yes	0
13	**TP with Non-Zero Routing ID Is Received** When a TP is received with a non-zero Tier 1 route string, this *Status* bit is Set. This bit is Cleared with a Hard Reset.	RO	Yes	0
14	**TP with Deferred Bit Set Is Received** When a TP is received with the *Deferred* bit set, this *Status* bit is Set. This bit is Cleared with a Hard Reset.	RO	Yes	0
31:15	***Reserved***	RsvdZ	Yes	0-0h

Register 16-28. 94h XCVRDIAG USB Transceiver Diagnostic Control (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
0	**Termination Select** 0 = High-Speed termination is enabled 1 = Full-Speed termination is enabled	RO	Yes	0
1	**Transceiver Select** 0 = High-Speed transceiver is enabled 1 = Full-Speed transceiver is enabled	RO	Yes	0
3:2	**Transceiver Operation Mode** Indicates the Transceiver operation mode (OPMODE). 00b = Normal 01b = Non-Driving 10b = Disable bit stuffing and NRZI encoding 11b = ***Reserved***	RO	Yes	00b
15:4	***Reserved***	RsvdZ	Yes	000h
17:16	**Line State** 00b = SE0 01b = J 10b = K 11b = SE1	PIN	Yes	00b
23:18	***Reserved***	RsvdZ	Yes	0-0h
26:24	**USB Test Mode** ***Valid only in High-Speed mode.*** 000b = Normal (default) 001b = Test J 010b = Test K 011b = Test SE0_NAK 100b = Test Packet 101b = Test Force Enable 110b, 111b = ***Reserved***	RW	Yes	000b
29:27	***Reserved***	RsvdZ	Yes	000b
30	**Force Full-Speed Mode** 1 = USB 3382 is forced into Full-Speed mode ***Note:* *Do not use this bit in standard operation*.** *It is provided for testing purposes only.*	RW	Yes	0
31	**Force High-Speed Mode** 1 = USB 3382 is forced into High-Speed mode ***Note:* *Do not use this bit in standard operation*.** *It is provided for testing purposes only.*	RW	Yes	0

Register 16-29. 98h SETUPDW0 Setup DWord 0 (USB Controller)

<table>
<tr><th>Bit(s)</th><th colspan="2">Description</th><th>Access</th><th>Serial EEPROM and I²C</th><th>Default</th></tr>
<tr><td rowspan="5">7:0</td><td colspan="2">Setup Byte 0
Provides Byte 0 of the last Setup packet received. For a Standard Device Request, returns the bmRequestType information listed below.</td><td rowspan="5">RO</td><td rowspan="5">Yes</td><td rowspan="5">00h</td></tr>
<tr><th>Bit(s)</th><th>Description/Function</th></tr>
<tr><td>4:0</td><td>Recipient
00h = Device
01h = Interface
02h = Endpoint
03h = Other
04h through 1Fh = Reserved</td></tr>
<tr><td>6:5</td><td>Type
00b = Standard
01b = Class
10b = Vendor
11b = Reserved</td></tr>
<tr><td>7</td><td>Direction
0 = Host to device
1 = Device to Host</td></tr>
<tr><td>15:8</td><td colspan="2">Setup Byte 1
Provides Byte 1 of the last Setup packet received. For a Standard Device Request, the bRequest code information, listed below, is returned.
00h = GET_STATUS (default)
01h = CLEAR_FEATURE
02h = Reserved
03h = SET_FEATURE
04h = Reserved
05h = SET_ADDRESS
06h = GET_DESCRIPTOR
07h = SET_DESCRIPTOR
08h = GET_CONFIGURATION
09h = SET_CONFIGURATION
0Ah = GET_INTERFACE
0Bh = GET_INTERFACE
0Ch = SYNCH_FRAME
30h = SET_SEL
31h = SET_ISOCH_DELAY</td><td>RO</td><td>Yes</td><td>00h</td></tr>
<tr><td>23:16</td><td colspan="2">Setup Byte 2
Provides Byte 2 of the last Setup packet received. For a Standard Device Request, the least significant byte (LSB) of the wValue field is returned.</td><td>RO</td><td>Yes</td><td>00h</td></tr>
<tr><td>31:24</td><td colspan="2">Setup Byte 3
Provides Byte 3 of the last Setup packet received. For a Standard Device Request, the most significant byte (MSB) of the wValue field is returned.</td><td>RO</td><td>Yes</td><td>00h</td></tr>
</table>

Register 16-30. 9Ch SETUPDW1 Setup DWord 1 (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
7:0	**Setup Byte 4** Provides Byte 4 of the last Setup packet received. For a Standard Device Request, the LSB of the *wIndex* field is returned.	RO	Yes	00h
15:8	**Setup Byte 5** Provides Byte 5 of the last Setup packet received. For a Standard Device Request, the MSB of the *wIndex* field is returned.	RO	Yes	00h
23:16	**Setup Byte 6** Provides Byte 6 of the last Setup packet received. For a Standard Device Request, the LSB of the *wIndex* field is returned.	RO	Yes	00h
31:24	**Setup Byte 7** Provides Byte 7 of the last Setup packet received. For a Standard Device Request, the MSB of the *wIndex* field is returned.	RO	Yes	00h

Register 16-31. A4h OURADDR Our USB Address (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
6:0	**Our USB Address** Contains the USB 3382's current USB address. This field is Cleared when a Root Port Reset (Host Port Reset) is detected. When written, the actual USB address is not changed until a valid Status stage is received from the USB Host.	RW	Yes	0-0h
7	**Force Immediate** If Set when this register is being written, the USB 3382's USB address is immediately updated, without waiting for a valid Status stage from the USB Host.	RW	Yes	0
31:8	***Reserved***	RsvdZ	Yes	0000_00h

Register 16-32. A8h OURCONFIG Our USB Configuration (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
7:0	**Our USB Configuration** Contains the USB 3382's current USB configuration. This field is Cleared when a Root Port Reset (Host Port Reset) is detected.	RW	Yes	00h
31:8	***Reserved***	RsvdZ	Yes	0000_00h

Register 16-33. B4h USB_CLASS USB Class, Sub-Class, Protocol (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
7:0	**Class** Determines the *bDeviceClass* returned in the Device Descriptor during an auto-enumeration.	RW	Yes	FFh
15:8	**Sub-Class** Determines the *bDeviceSubClass* returned in the Device Descriptor during an auto-enumeration.	RW	Yes	00h
23:16	**Protocol** Determines the *bDeviceProtocol* returned in the Device Descriptor during an auto-enumeration.	RW	Yes	00h
31:24	***Reserved***	RsvdZ	Yes	00h

Register 16-34. B8h SS_SEL SuperSpeed System Exit Latency (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
7:0	**U1 System Exit Latency** Total latency to transition the entire paths of Links between the USB 3382 and Host, from the U1 to U0 Link state, under worst case circumstances, when the USB 3382 initiates the transition. 00h = Zero 01h = Less than 1 µs 02h = Less than 2 µs 03h = Less than 3 µs 04h = Less than 4 µs … 0Ah = Less than 10 µs 0Bh – FFh = ***Reserved***	RW	Yes	00h
23:8	**U2 System Exit Latency** Total latency to transition the entire paths of Links between the USB 3382 and Host, from the U2 to U0 Link state, under worst case circumstances, when the USB 3382 initiates the transition. 0000h = Zero 0001h = Less than 1 µs 0002h = Less than 2 µs 0003h = Less than 3 µs 0004h = Less than 4 µs … 7FFh = Less than 2,047 µs 800h – FFFFh = ***Reserved***	RW	Yes	0000h
31:24	***Reserved***	RsvdZ	Yes	00h

Register 16-35. BCh SS_DEL SuperSpeed Device Exit Latency (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
7:0	**U1 Device Exit Latency** Worst case latency to transition from the U1 to U0 Link state, assuming that the latency is limited only by the USB 3382, and not the USB 3382's Link partner. 00h = Zero 01h = Less than 1 µs 02h = Less than 2 µs 03h = Less than 3 µs 04h = Less than 4 µs … 0Ah = Less than 10 µs 0Bh – FF = ***Reserved***	RW	Yes	04h
23:8	**U2 Device Exit Latency** Worst case latency to transition from the U2 to U0 Link state, assuming that the latency is limited only by the USB 3382, and not the USB 3382's Link partner. 0000h = Zero 0001h = Less than 1 µs 0002h = Less than 2 µs 0003h = Less than 3 µs 0004h = Less than 4 µs … 7FFh = Less than 2,047 µs 800h – FFFFh = ***Reserved***	RW	Yes	0041h
31:24	***Reserved***	RsvdZ	Yes	00h

Register 16-36. C0h USB2LPM USB2 Link Power Management (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**USB L1 LPM Support** This bit is returned in the *USB r2.0* Extension Descriptor *Link Power Management (LPM)* bit, bit 1, and indicates that the USB 3382 supports the Link Power Management protocol.	RW	Yes	1
1	**USB L1 LPM Remote Wake Enable** This bit is Set/Cleared by bit 8 of an LPM Extended Transaction. It is used to enable Device Remote Wakeup in *USB r2.0* mode.	RO	Yes	0
5:2	**USB L1 LPM Hird** The Host-Initiated Resume Duration (Hird) indicates to the USB 3382 how long the Host will drive the Resume, when the Host initiates exit from the L1 Link PM state. The USB 3382 uses the Hird value to help determine which Power Optimization features it should use in response. Set by bits [7:4] of an LPM Extended Transaction.	RO	Yes	0h
31:6	***Reserved***	RsvdZ	Yes	0-0h

Register 16-37. C4h USB3BELT USB3 Best Effort Latency Tolerance (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
9:0	**Best Effort Latency Tolerance (BELT)** Represents the length of time (in nanoseconds) that the USB 3382 can wait for service before experiencing unintended operation side effects. The BELT value can be modified by the multiplier indicated in field [11:10] (*BELT Multiplier*).	RW	Yes	1Fh
11:10	**BELT Multiplier** Provides a multiplier for field [9:0] (*Best Effort Latency Tolerance (BELT)*). 00b = ***Reserved*** 01b = 2^{10} (1,024) 10b = 2^{15} (32,768) 11b = 2^{20} (1,048,576)	RO	Yes	10b
31:12	***Reserved***	RsvdZ	Yes	0000_0h

Register 16-38. C8h USBCTL2 USB Control 2 (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**Serial Number String Enable** 0 = *Serial String Index* value in the Device Descriptor is Cleared, and the string Descriptor Read is acknowledged with a stall in Auto-Enumerate mode 1 = Allows the USB 3382 serial number String Descriptor to be returned to the Host in Auto-Enumerate mode	RW	Yes	0
1	***Reserved***	RsvdZ	Yes	0
2	**USB2 Core Enable** 1 = Enables the *USB r2.0* core to begin speed negotiations with the Host	RW	Yes	1
3	**USB3 Core Enable** 1 = Enables the *USB r3.0* core to begin Link training with the Host	RW	Yes	1
4	**Function Suspend** This bit can be Set or Cleared with a USB Standard Request, as well as a direct register Write. 1 = USB 3382 is in a Suspended state	RW	Yes	0
5	**U1 Enable** This bit can be Set or Cleared with a USB Standard Request, as well as a direct register Write. 1 = Allows the *USB r3.0* Link to go to the U1 Link state	RW	Yes	0
6	**U2 Enable** This bit can be Set or Cleared with a USB Standard Request, as well as a direct register Write. 1 = Allows the *USB r3.0* Link to go to the U2 Link state	RW	Yes	0
7	**LTM Enable** Used in conjunction with the **USB3BELT** register *Best Effort Latency Tolerance (BELT)* field (USB Controller, offset C4h[9:0]). (Refer to the *USB r3.0*, Section 8.5.6.5, for details.) This bit can be Set or Cleared with a USB Standard Request, as well as a direct register Write. 1 = Allows the USB 3382 to generate Latency Tolerance Messaging (LTM) Messages	RW	Yes	0
31:8	***Reserved***	RsvdZ	Yes	0000_00h

Register 16-39. CCh IN_TIMEOUT IN Endpoint Credit Timeout (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**GPEP0 Timeout Enable** 1 = Timeout timer runs when GPEP0 IN Packet data is not making progress, and data is not being received from the PCI Express interface	RW	Yes	0
3:1	**GPEP0 Timeout Value** 000b = 16 ms 001b = 32 ms 010b = 64 ms 011b = 1,024 ms All other encodings are ***reserved***.	RW	Yes	000b
4	**GPEP1 Timeout Enable** 1 = Timeout timer runs when GPEP1 IN Packet data is not making progress, and data is not being received from the PCI Express interface	RW	Yes	0
7:5	**GPEP1 Timeout Value** 000b = 16 ms 001b = 32 ms 010b = 64 ms 011b = 1,024 ms All other encodings are ***reserved***.	RW	Yes	000b
8	**GPEP2 Timeout Enable** 1 = Timeout timer runs when GPEP2 IN Packet data is not making progress, and data is not being received from the PCI Express interface	RW	Yes	0
11:9	**GPEP2 Timeout Value** 000b = 16 ms 001b = 32 ms 010b = 64 ms 011b = 1,024 ms All other encodings are ***reserved***.	RW	Yes	000b
12	**GPEP3 Timeout Enable** 1 = A timeout timer runs when GPEP3 IN Packet data is not making progress, and data is not being received from the PCI Express interface	RW	Yes	0
15:13	**GPEP3 Timeout Value** 000b = 16 ms 001b = 32 ms 010b = 64 ms 011b = 1,024 ms All other encodings are ***reserved***.	RW	Yes	000b

Register 16-39. CCh IN_TIMEOUT IN Endpoint Credit Timeout (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
16	**GPEP0 Timeout** Writing 1 Clears the bit. 1 = GPEP0 timeout occurred	RW1C	Yes	0
17	**GPEP1 Timeout** Writing 1 Clears the bit. 1 = GPEP1 timeout occurred	RW1C	Yes	0
18	**GPEP2 Timeout** Writing 1 Clears the bit. 1 = GPEP2 timeout occurred	RW1C	Yes	0
19	**GPEP3 Timeout** Writing 1 Clears the bit. 1 = GPEP3 timeout occurred	RW1C	Yes	0
31:20	***Reserved***	RsvdZ	No	000h

Register 16-40. D0h ISODELAY Isochronous Delay (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
15:0	**Isochronous Delay** This delay represents the time from when the Host starts transmitting the first framing symbol of the packet, to when the device receives the first framing symbol of that packet. The range is 0 to 65,535 ns.	RW	Yes	0000h
31:16	***Reserved***	RsvdZ	No	0000h

16.7 PCI Express/Configuration Cursor Registers

Note: *USB accesses to the PCIOUT, PCIIN, CSROUT, and CSRIN Dedicated endpoints modify the values in these registers.*

Table 16-6. PCI Express/Configuration Cursor Registers

Offset (from BAR0)	Register	Description
100h	PCIMSTCTL	PCI Master Control
104h	PCIMSTADDR	PCI Master Address
108h	PCIMSTDATA	PCI Master Data
10Ch	PCIMSTSTAT	PCI Express Master Status
110h	CSRCTL	CSR Control
114h	CSRDATA	CSR Data
118h	SEMAPHORE	General Semaphore
11Ch	PCIMSTMSG	PCI Master Message
120h – 17Fh	***Reserved***	

Register 16-41. 100h PCIMSTCTL PCI Master Control (USB Controller)

<table>
<tr><th>Bit(s)</th><th>Description</th><th>Access</th><th>Serial EEPROM and I^2C</th><th>Default</th></tr>
<tr><td>3:0</td><td>PCI Express First Byte Enables
Determines the first Byte Enables of a PCI Express transaction. For 1-DWord transactions, it can be any value. For multiple DWord transactions, only contiguous Byte Enables are allowed, or the endpoint is halted. This field is used directly in the FBE field of the PCI Express Header.</td><td>RW</td><td>Yes</td><td>0h</td></tr>
<tr><td>5:4</td><td>PCI Express Master Command Select
When the USB 3382 performs PCI Express transactions initiated by the PCIOUT endpoint or 8051, determines the PCI Express Request type issued.
Note: The Configuration Type (Type 0 or Type 1) is determined by the PCI Master Address format.
<table>
<tr><th>Value</th><th>Read Command</th><th>Write Command</th></tr>
<tr><td>00b</td><td>Memory Read</td><td>Memory Write</td></tr>
<tr><td>01b</td><td>I/O Read</td><td>I/O Write</td></tr>
<tr><td>10b</td><td>Configuration Read</td><td>Configuration Write</td></tr>
<tr><td>11b</td><td>Reserved</td><td>PCI Express Message</td></tr>
</table></td><td>RW</td><td>Yes</td><td>00b</td></tr>
<tr><td>6</td><td>PCI Express Master Start
Writing 1 causes a PCI Write or Read transaction to start. This bit is Cleared when the PCI transaction is complete.
For Write operations, determines when to start another Write.
For Read operations, determines when the PCIMSTDATA register (USB Controller, offset 108h) contains valid data.
This bit is automatically Cleared when a UR or CA occurs.</td><td>RW1S</td><td>Yes</td><td>0</td></tr>
<tr><td>7</td><td>PCI Express Master Read/Write
0 = PCI Write transaction is selected.
1 = PCI Read transaction is selected. For 8051 Writes to the PCI Express interface, this bit must be Cleared before the PCIMSTDATA register (USB Controller, offset 108h) is written.</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td>15:8</td><td>Message Code
This field defines the Message code, when the Command Select is PCI Express Message (field [5:4] is programmed to 11b, for a Write Command).</td><td>RW</td><td>Yes</td><td>00h</td></tr>
<tr><td>18:16</td><td>Message Routing
Determines the Message routing mechanism used. A value of 011b indicates a Broadcast Message, and is not supported by the USB 3382.</td><td>RW</td><td>Yes</td><td>000b</td></tr>
<tr><td>19</td><td>Message Type
0 = Msg
1 = MsgD</td><td>RW</td><td>Yes</td><td>0</td></tr>
</table>

Register 16-41. 100h PCIMSTCTL PCI Master Control (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
20	**PCI Express Port** Determines the destination PCI Express Port. 0 = Port 0 1 = Port 1	RW	Yes	0
23:21	***Reserved***	RsvdZ	Yes	000b
30:24	**PCI Express DW Length** Determines the DWord length of the PCI Express Write or Read Request. The maximum PCI Express TLP length is 64 DWords for PCIOUT transactions. This field is used directly in the *Length* field of the PCI Express Header. For 8051 PCI Express Writes, this field must always have a value of 01h.	RW	Yes	0-0h
31	***Reserved***	RsvdZ	Yes	0h

Register 16-42. 104h PCIMSTADDR PCI Master Address (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
31:0	**PCI Master Address** Determines the PCI Express Master transaction target PCI DWord address. Bits [1:0] are ignored for Memory and I/O transactions, and are used only for Configuration and Message transactions.	RW	Yes	0000_0000h

Register 16-43. 108h PCIMSTDATA PCI Master Data (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
31:0	**PCI Master Data** Data being transferred to and/or from the PCI Express interface is accessed through this register. For PCI Express Writes, the **PCIMSTCTL** register *PCI Express Master Read/ Write* bit (USB Controller, offset 100h[7]) must be Cleared before this register is written. For PCI Express Reads, this register is valid when the **PCIMSTCTL** register *PCI Express Master Start* bit (USB Controller, offset 100h[6]) is Cleared.	RW	Yes	0000_0000h

Register 16-44. 10Ch PCIMSTSTAT PCI Express Master Status (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
0	**Root Complex Mode** Reflects the STRAP_RC_MODE input status. 0 = USB 3382 operates in Adapter mode (STRAP_RC_MODE is pulled or tied Low to Ground) 1 = USB 3382 operates in Root Complex mode (STRAP_RC_MODE is pulled High to VDD_IO)	PIN	Yes	0
1	***Reserved***	RsvdZ	Yes	0
31:2	***Reserved***	RsvdZ	Yes	0-0h

Register 16-45. 110h CSRCTL CSR Control (USB Controller)

<table>
<tr><th>Bit(s)</th><th>Description</th><th>Access</th><th>Serial EEPROM and I²C</th><th>Default</th></tr>
<tr><td colspan="5">This register is automatically written when a USB packet arrives in the CSROUT endpoint.</td></tr>
<tr><td>3:0</td><td>CSR Byte Enables
Determines which bytes of the CSRDATA register (USB Controller, offset 114h) are applied to the CSR Write.<table><tr><th>Bit</th><th>Byte</th></tr><tr><td>0</td><td>CSRDATA[7:0] (LSB)</td></tr><tr><td>1</td><td>CSRDATA[15:8]</td></tr><tr><td>2</td><td>CSRDATA[23:16]</td></tr><tr><td>3</td><td>CSRDATA[31:24] (MSB)</td></tr></table></td><td>RW</td><td>Yes</td><td>0h</td></tr>
<tr><td>5:4</td><td>CSR Space Select
Determines which Address space is accessed.
00b = PCI Express Configuration registers
01b = Memory-Mapped Configuration registers
10b = 8051 Program RAM
11b = Reserved</td><td>RW</td><td>Yes</td><td>00b</td></tr>
<tr><td>6</td><td>CSR Start
Writing 1 causes a Configuration Write or Read transaction to start. Cleared when the CSR transaction is complete.
For Write operations, determines when to start another Write.
For Read operations, determines when the CSRDATA register (USB Controller, offset 114h) contains valid data.</td><td>RW1S</td><td>Yes</td><td>0</td></tr>
<tr><td>7</td><td>CSR Read/Write
0 = CSR Write transaction is selected
1 = CSR Read transaction is selected</td><td>RW</td><td>Yes</td><td>0</td></tr>
<tr><td>15:8</td><td>Reserved</td><td>RsvdZ</td><td>Yes</td><td>00h</td></tr>
<tr><td>31:16</td><td>CSR Address
Determines the CSR transaction’s Destination address.
Note: Address must be DWord-aligned (bits [17:16] must be Cleared).</td><td>RW</td><td>Yes</td><td>0000h</td></tr>
</table>

Register 16-46. 114h CSRDATA CSR Data (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
31:0	**CSR Data** Data being transferred to and/or from the CSR, whose address is contained in the **CSRCTL** register *CSR Address* field (USB Controller, offset 110h[31:16]), is accessed through this register. For Reads, this register is valid when the **CSRCTL** register *CSR Start* bit (USB Controller, offset 110h[6]) is Cleared.	RW	Yes	0000_0000h

Register 16-47. 118h SEMAPHORE General Semaphore (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
0	**Semaphore** This bit can be used by the USB Host and 8051, to determine which one acquires access to the on-chip CSRs. When this bit is read, the current value is returned, and the bit is automatically Set. Reading a value of 0 indicates that the device (USB Host or 8051) has permission to access CSRs. When the controlling device is finished using the resource, it writes a 0 to Clear the bit.	RWC	Yes	0
31:1	***Reserved***	RsvdZ	Yes	0-0h

Register 16-48. 11Ch PCIMSTMSG PCI Master Message (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
31:0	**PCI Master Message** When the USUSB Host or 8051 sends a PCI Express Message, this register provides the 4th DWord of the Message Header. The other three DWords of the Message Header are derived from the **PCIMSTCTL** register (USB Controller, offset 100h).	RW	Yes	0000_0000h

16.8 DMA Registers

Table 16-7. DMA Registers

Register	Offset (from BAR0)			
	Channel 0	Channel 1	Channel 2	Channel 3
DMACTL	180h	1A0h	1C0h	1E0h
DMASTAT	184h	1A4h	1C4h	1E4h
Reserved	188h – 18Ch	1A8h – 1ACh	1C8h – 1CCh	1E8h – 1ECh
DMACOUNT	190h	1B0h	1D0h	1F0h
DMAADDR	194h	1B4h	1D4h	1F4h
DMADESC	198h	1B8h	1D8h	1F8h
Reserved	19Ch	1BCh	1DCh	1FCh
DMACOUNTP	680h	690h	6A0h	6B0h
DMAADDRP	684h	694h	6A4h	6B4h
DMADESCP	688h	698h	6A8h	6B8h
Reserved	68Ch	69Ch	6ACh	6BCh

Register 16-49. 180h, 1A0h, 1C0h, 1E0h DMACTL DMA Control (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**DMA Address Constant** 0 = PCI Express address is incremented for each successive PCI Express TLP 1 = Same PCI Express address is used for all TLPs issued during the DMA transfer	RW	Yes	0
1	**DMA Enable** If a DMA transfer is in progress when this bit is Cleared, the Data transfer and Descriptor processing are paused. This bit is automatically Cleared when the channel's **DMASTAT** register *DMA Abort* bit(s) (USB Controller, offset(s) 184h, 1A4h, 1C4h, 1E4h[1]), or a PCI Master or Target Abort occurs during a DMA transaction. 0 = DMA channel is disabled. Value read back is 1, until the DMA channel completes the Pause sequence. 1 = DMA channel is enabled to transfer data and process DMA Descriptors.	RWU	Yes	0
2	**DMA FIFO Validate** ***Valid only for Single or Descriptor DMA transfers.*** 0 = Last short packet is not automatically validated at the end of a DMA transfer. Auto-validation can be enabled during Descriptor transfers, by way of the channel's **DMACOUNT** register *DMA Descriptor FIFO Validate* bit(s) (USB Controller, offset(s) 190h, 1B0h, 1D0h, 1F0h[27]). 1 = Last short packet is automatically validated at the end of a DMA transfer.	RW	Yes	0
3	***Reserved***	RW	Yes	0
4	**DMA OUT Auto Start Enable** 1 = DMA channel automatically starts when an OUT packet is received	RW	Yes	0
7:5	**DMA Requests Outstanding** Determines the quantity of outstanding Write or Read Requests that a DMA channel can send, before a Request is de-allocated.	RW	Yes	110b
15:8	***Reserved***	RsvdP	No	00h

Register 16-49. 180h, 1A0h, 1C0h, 1E0h DMACTL DMA Control (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
16	**DMA Descriptor Mode** 0 = Block mode. A single DMA operation is performed when the channel's **DMASTAT** register *DMA Start* bit(s) (USB Controller, offset(s) 184h, 1A4h, 1C4h, 1E4h[0]) is Set. The channel's **DMACOUNT** (USB Controller, offset(s) 190h, 1B0h, 1D0h, 1F0h) and **DMAADDR** (USB Controller, offset(s) 194h, 1B4h, 1D4h, 1F4h) registers define the transfer. 1 = Descriptor mode. The DMA channel traverses a Linked list of Descriptors, each of which defines a DMA operation. The list of Descriptors is located in either off-chip PCI Express System memory or on-chip 8051 memory.	RW	Yes	0
17	**DMA Valid Bit Enable** 0 = DMA Descriptors are processed without regard to the channel's **DMACOUNT** register *Valid Bit* bit(s) (USB Controller, offset(s) 190h, 1B0h, 1D0h, 1F0h[31]). 1 = DMA Descriptors are processed only when the channel's *Valid Bit* bit(s) is Set. If a valid Descriptor with a 0-Byte Count is encountered for an: • OUT endpoint, the USB 3382 moves on to the next Descriptor • IN endpoint, and the channel's **DMACTL** register *DMA FIFO Validate* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[2]) is not Set, the USB 3382 moves on to the next Descriptor • IN endpoint, and the channel's *DMA FIFO Validate* bit(s) is Set, then a Zero-Length packet is written to the endpoint FIFO, before the USB 3382 moves on to the next Descriptor	RW	Yes	0
18	**DMA Valid Bit Polling Enable** ***Valid only if bits [17 and 16] (DMA Valid Bit Enable and DMA Descriptor Mode, respectively) for the channel are both Set.*** 0 = USB 3382 stops polling for valid Descriptors, if it encounters a channel's **DMACOUNT** register *Valid Bit* bit(s) (USB Controller, offset(s) 190h, 1B0h, 1D0h, 1F0h[31]) that is not Set 1 = USB 3382 continues polling the same Descriptor, if it encounters a channel's *Valid Bit* bit that is not Set	RW	Yes	0
20:19	**Descriptor Polling Rate** Selects the polling rate when a channel's Cleared **DMACOUNT** register *Valid Bit* bit(s) (USB Controller, offset(s) 190h, 1B0h, 1D0h, 1F0h[31]) is detected. 00b = Continuous 01b = 1 µs 10b = 100 µs 11b = 1 ms	RW	Yes	01b

Register 16-49. 180h, 1A0h, 1C0h, 1E0h DMACTL DMA Control (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
21	**DMA Clear Count Enable** 0 = DMA Descriptor *Byte Counter* field is not changed at the end of a transfer 1 = Channel's **DMACOUNT** register *Valid Bit* bit(s) (USB Controller, offset(s) 190h, 1B0h, 1D0h, 1F0h[31]) is Cleared, and the current DMA Count is written to the same register's *DMA Transfer Length* field (field [23:0]) after the DMA transfer completes	RW	Yes	0
22	**Prefetch Disable** 0 = Descriptor Prefetch registers are loaded from memory. 1 = Descriptor Prefetch registers are not loaded from memory. In this case, the USB Host writes the Prefetch registers, using the CFGOUT endpoint.	RW	Yes	0
23	**Pause Mode** 0 = Immediate Pause mode. All PCI Express data and Descriptor Requests are stopped when the channel's **DMACTL** register *DMA Enable* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[1]) is Cleared. 1 = Graceful Pause mode. The current Descriptor is allowed to finish; however, no new Descriptors are read when the channel's *DMA Enable* bit(s) is Cleared.	RW	Yes	0
24	***Reserved***	RsvdP	No	0
25	**DMA Descriptor Done Interrupt Enable** 1 = An interrupt generates when the last Descriptor in the Descriptor Linked list completes its transfer (channel's **DMACOUNT** register *Last Descriptor* bit(s) (USB Controller, offset(s) 190h, 1B0h, 1D0h, 1F0h[28]) is Set)	RW	Yes	0
26	**DMA Pause Done Interrupt Enable** 1 = An interrupt generates when the DMA Channel Pause sequence completes	RW	Yes	0
27	**DMA Abort Done Interrupt Enable** 1 = An interrupt generates when the DMA Channel Abort sequence completes	RW	Yes	0
31:28	***Reserved***	RsvdP	No	0h

Register 16-50. 184h, 1A4h, 1C4h, 1E4h DMASTAT DMA Status (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**DMA Start** Writing 1 causes the DMA channel to start. This bit is self-Clearing, and always returns a value of 0.	W1RZ	Yes	0
1	**DMA Abort** Writing 1 causes the DMA transfer to abort. When read, this bit remains Set until the Abort sequence completes.	W1R	Yes	0
23:2	***Reserved***	RO	No	0-0h
24	**DMA Transaction Done Interrupt** An OUT DMA is done when its Counter reaches 0, or when a short or ZLP OUT packet is received and the last PCI Express Write TLP has been ACKed. An IN DMA is done when its Counter reaches 0 and all partial Completions have been received by the USB 3382. Writing 1 Clears this status bit. 1 = Current DMA in progress completes its transfer, and the channel's **DMACOUNT** register *DMA Done Interrupt Enable* bit(s) (USB Controller, offset(s) 190h, 1B0h, 1D0h, 1F0h[29]) is Set.	RW1C	Yes	0
25	**DMA Last Descriptor Done Interrupt** 1 = Last Descriptor in the Descriptor Linked list completed its transfer. Writing 1 Clears this status bit.	RW1C	Yes	0
26	**DMA Pause Done Interrupt** Writing 1 Clears this status bit. 1 = DMA Channel Pause sequence completed. The Pause sequence is initiated by Clearing the channel's **DMACTL** register *DMA Enable* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[1]).	RW1C	Yes	0
27	**DMA Abort Done Interrupt** Writing 1 Clears this status bit. 1 = DMA Channel Abort sequence completed. The Abort sequence is initiated by Setting the channel's **DMASTAT** register *DMA Abort* bit(s) (USB Controller, offset(s) 184h, 1A4h, 1C4h, 1E4h[1]).	RW1C	Yes	0
30:28	***Reserved***	Rsvd	No	000b
31	**DMA Completion Sequence Error Status**	RW1C	Yes	0

Register 16-51. 190h, 1B0h, 1D0h, 1F0h DMACOUNT DMA Transfer Length (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
23:0	**DMA Transfer Length** Determines the total quantity of bytes to be transferred. The maximum DMA transfer size is 16 MB. This register is decremented as data is transferred.	RWU	Yes	00_0000h
24	**DMA OUT Continue** 0 = When a short OUT packet is received, the endpoint's **EP_STAT** register *NAK Packets* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[4]) is Set, if the endpoint's **EP_RSP** register *NAK OUT Packets Mode Set* and *NAK OUT Packets Mode Clear* bit(s) (USB Controller, offset(s) 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h[10 and 2], respectively) are also Set. Also, the Descriptor DMA channel stops if the DMA Counter did not reach 0. 1 = When a short OUT packet is received, the endpoint's *NAK Packets* bit(s) is Set, but is automatically Cleared when the DMA channel finishes transferring the short packet to the PCI Express interface. Also, the Descriptor Controller is then enabled to read the next Descriptor. When this bit is Set, the endpoint's *NAK OUT Packets Mode Set* and *NAK OUT Packets Mode Clear* bit(s) must also be Set.	RW	Yes	0
26:25	***Reserved***	RO	No	00b
27	**DMA Descriptor FIFO Validate** 0 = Last short packet is not automatically validated at the end of a DMA transfer. 1 = Last short packet is automatically validated at the end of a DMA transfer. The channel's **DMACTL** register *DMA FIFO Validate* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[2]) takes precedence over this bit. If the *DMA FIFO Validate* bit is Set, all USB short packets are automatically validated at the end of each Descriptor DMA transfer.	RW	Yes	0
28	**Last Descriptor** 0 = There are additional DMA Descriptors following the current Descriptor being processed 1 = This Descriptor is the last Descriptor in the Descriptor Linked list	RWU	Yes	0
29	**DMA Done Interrupt Enable** 1 = An interrupt is generated when field [23:0] (*DMA Transfer Length*) for this Descriptor reaches 0	RWU	Yes	0
30	**DMA Direction** 0 = DMA channel transfers data from the USB to PCI Express interface for OUT endpoints 1 = DMA channel transfers data from the PCI Express to USB interface for IN endpoints	RWU	Yes	0
31	**Valid Bit** Reflects the state of the *Valid* bit in the current DMA Descriptor. 0 = Not Valid 1 = Valid	RWU	Yes	0

Register 16-52. 194h, 1B4h, 1D4h, 1F4h DMAADDR DMA Address (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
31:0	**DMA Address** Determines the PCI Express starting address of a DMA transfer. DMA starting addresses can be aligned on any Byte boundary. The DMA address might be incremented, depending upon the channel's **DMACTL** register *DMA Address Constant* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[0]).	RWU	Yes	0000_0000h

Register 16-53. 198h, 1B8h, 1D8h, 1F8h DMADESC DMA Descriptor (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
0	**On-Chip** 0 = Descriptors are stored in off-chip PCI Express memory 1 = Descriptors are stored in on-chip 8051 memory	RW	Yes	0
1	**Descriptor Port Select** 0 = Off-chip Descriptor Requests are sent to PCI Express Port 0 1 = Off-chip Descriptor Requests are sent to PCI Express Port 1	RW	Yes	0
2	**Data Port Select** 0 = DMA channel Data Requests are sent to PCI Express Port 0 1 = DMA channel Data Requests are sent to PCI Express Port 1	RW	Yes	0
3	***Reserved***	RsvdP	No	0
31:4	**Next Descriptor Address** Points to the next DMA Descriptor to be processed.	RWU	Yes	0000_000h

Register 16-54. 680h, 690h, 6A0h, 6B0h DMACOUNTP DMA Transfer Length (Prefetch) (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
23:0	**DMA Transfer Length** Determines the total quantity of bytes to be transferred. The maximum DMA transfer size is 16 MB. This register is decremented as data is transferred.	RWU	Yes	00_0000h
24	**DMA OUT Continue** 0 = When a short OUT packet is received, the endpoint's **EP_STAT** register *NAK Packets* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[4]) is Set, if the endpoint's **EP_RSP** register *NAK OUT Packets Mode Set* and *NAK OUT Packets Mode Clear* bit(s) (USB Controller, offset(s) 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h[10 and 2], respectively) are also Set. Also, the Descriptor DMA channel stops if the DMA Counter did not reach 0. 1 = When a short OUT packet is received, the endpoint's *NAK Packets* bit(s) is Set, but is automatically Cleared when the DMA channel finishes transferring the short packet to the PCI Express interface. Also, the Descriptor Controller is then enabled to read the next Descriptor.	RWU	Yes	0
26:25	**DMA ISO Extra Transaction Opportunity** ***Note:*** *These bits are loaded from the DMACOUNT word in the prefetched DMA Descriptor to the endpoint's* ***GPEP[3:0/Out/In]_HS_MAXPKT*** *register Additional Transaction Opportunities field(s) (USB Controller, Index 20h, 30h, 40h, 50h, 60h, 70h, 80h, 90h[12:11]).*	RWU	Yes	00b
27	**DMA Descriptor FIFO Validate** 0 = Last short packet is not automatically validated at the end of a DMA transfer. 1 = Last short packet is automatically validated at the end of a DMA transfer. The channel's **DMACTL** register *DMA FIFO Validate* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[2]) takes precedence over this bit. If the *DMA FIFO Validate* bit is Set, all USB short packets are automatically validated at the end of each Descriptor DMA transfer.	RWU	Yes	0
28	**Last Descriptor** 0 = There are additional DMA Descriptors following the current Descriptor being processed 1 = This Descriptor is the last Descriptor in the Descriptor Linked list	RWU	Yes	0
29	**DMA Done Interrupt Enable** 1 = An interrupt is generated when field [23:0] (*DMA Transfer Length*) for this Descriptor reaches 0	RWU	Yes	0
30	**DMA Direction** 0 = DMA channel transfers data from the USB to PCI Express interface for OUT endpoints 1 = DMA channel transfers data from the PCI Express to USB interface for IN endpoints	RWU	Yes	0
31	**Valid Bit** Reflects the state of the *Valid* bit in the prefetched DMA Descriptor. 0 = Not Valid 1 = Valid	RWU	Yes	0

Register 16-55. 684h, 694h, 6A4h, 6B4h DMAADDRP DMA Address (Prefetch) (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
31:0	**DMA Address** Determines the PCI Express starting address of a DMA transfer. DMA starting addresses can be aligned on any Byte boundary. The DMA address might be incremented, depending upon the channel's **DMACTL** register *DMA Address Constant* bit(s) (USB Controller, offset(s) 180h, 1A0h, 1C0h, 1E0h[0]).	RWU	Yes	0000_0000h

Register 16-56. 688h, 698h, 6A8h, 6B8h DMADESCP DMA Descriptor (Prefetch) (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**On-Chip** 0 = Descriptors are stored in off-chip PCI Express memory 1 = Descriptors are stored in on-chip DMA Descriptor memory	RW	Yes	0
1	**Descriptor Port Select** 0 = Off-chip Descriptor Requests are sent to PCI Express Port 0 1 = Off-chip Descriptor Requests are sent to PCI Express Port 1	RW	Yes	0
2	**Data Port Select** 0 = DMA channel Data Requests are sent to PCI Express Port 0 1 = DMA channel Data Requests are sent to PCI Express Port 1	RW	Yes	0
3	***Reserved***	RsvdP	No	0
31:4	**Next Descriptor Address** Points to the next DMA Descriptor to be processed.	RWU	Yes	0000_000h

16.9 Dedicated Endpoint Registers

Table 16-8. Dedicated Endpoint Registers

Register	Offset (from BAR0)					
	CSROUT	CSRIN	PCIOUT	PCIIN	STATIN	RCIN
DEP_CFG	200h	210h	220h	230h	240h	250h
DEP_RSP	204h	214h	224h	234h	244h	254h
Reserved	208h – 20Ch, 218h – 21Ch, 228h – 22Ch, 238h – 23Ch, 248h – 24Ch					

Register 16-57. 200h, 210h, 220h, 230h, 240h, 250h DEP_CFG Dedicated Endpoint Configuration for CSROUT, CSRIN, PCIOUT, PCIIN, STATIN, and RCIN (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
3:0	**Endpoint Number** Selects the endpoint number.	RW	Yes	RCIN = Ch, CSROUT = Dh, CSRIN = Dh, PCIOUT = Eh, PCIIN = Eh, STATIN = Fh
7:4	***Reserved***	RsvdZ	Yes	0h
8	**Endpoint Type** 0 = STATIN or RCIN endpoint becomes a BULK endpoint. 1 = STATIN or RCIN endpoint becomes an INTERRUPT endpoint. Valid only for the STATIN or RCIN endpoint. All other endpoints are BULK.	RW	Yes	STATIN = 1, RCIN = 1, Others = 0
9	***Reserved***	RsvdZ	Yes	0
10	**Endpoint Enable** 1 = Enables this endpoint	RW	Yes	RCIN = 0 in Adapter mode, Others = 1
15:11	**Service Interval** Determines the interrupt service interval for STATIN/RCIN endpoints in *USB r3.0* mode.	RW	Yes	STATIN = 1, RCIN = 1, Others = 0
31:16	***Reserved***	RsvdZ	Yes	0000h

Register 16-58. 204h, 214h, 224h, 234h, 244h, 254h DEP_RSP Dedicated Endpoint Response for CSROUT, CSRIN, PCIOUT, PCIIN, STATIN, and RCIN (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
Note: *Writing 1 to bits [7, 1:0] of this register Clears the corresponding register bits. Writing 1 to bits [15, 9:8] Sets the corresponding register bits. When this register is read, bits [7, 1:0] are duplicated on bits [15, 9:8].*				
0	**Endpoint Halt Clear** Clears the *Endpoint Halt* bit. When an Endpoint Clear Feature Standard Request to the *Endpoint Halt* bit is detected by the CPU, the CPU must write 1 to this bit. Reading this bit returns the current state of the *Endpoint Halt* bit.	RW1C	Yes	0
1	**Endpoint Toggle Clear** Clears the *Endpoint Toggle* bit. Reading this bit returns the current *Endpoint Toggle* bit state. The *Endpoint Toggle* bit automatically changes its state after each *USB r2.0* packet is successfully transferred. The CPU Clears the *Endpoint Toggle* bit only for Endpoint initialization events. Reading this bit returns the current state of the endpoint *Data* toggle bit.	RW1C	Yes	0
2	**Endpoint FIFO Flush** Writing this bit flushes the endpoint's FIFO. Reading this bit always returns a value of 0.	RW1C	Yes	0
6:3	***Reserved***	RsvdZ	Yes	0h
7	**NAK Packets Clear** **OUT Endpoints** 1 = All OUT or PING packets cause a NAK/NRDY handshake to return to the USB Host **In Endpoints** 1 = All IN tokens cause a NAK/NRDY handshake to return to the USB Host	RW1C	Yes	0
8	**Endpoint Halt Set** Refer to Note at beginning of this register.	RW1S	Yes	0
9	**Endpoint Toggle Set** Refer to Note at beginning of this register.	RW1S	Yes	0
10	**Endpoint FIFO Flush 2** Writing this bit flushes the endpoint's FIFO. Reading this bit always returns a value of 0.	RW1C	Yes	0
14:11	***Reserved***	RsvdZ	Yes	0h
15	**NAK Packets Set** Refer to Note at beginning of this register.	RW1S	Yes	0
31:16	***Reserved***	RsvdZ	Yes	0000h

16.10 EP 0 and GPEP*x* Registers

There is one **EP_CFG** register for EP 0, and one **EP_CFG** and **EP_VAL** register for each pair of General-Purpose Endpoints. For the remainder of the registers, there is one for EP 0 and one for each direction (IN and OUT) of the four General-Purpose endpoints.

Note: *In Legacy Adapter mode, only the "OUT" register offset is used to access the corresponding register, although the endpoint can be programmed as IN or OUT.*

Table 16-9. EP 0 and GPEP*x* Registers

Register	Offset (from BAR0)				
	EP 0	GPEP0 OUT/IN	GPEP1 OUT/IN	GPEP2 OUT/IN	GPEP3 OUT/IN
EP_CFG	300h	320h	340h	360h	380h
EP_RSP	304h	324h/3E4h	344h/404h	364h/424h	384h/444h
EP_IRQENB	308h	328h/3E8h	348h/408h	368h/428h	388h/448h
EP_STAT	30Ch	32Ch/3ECh	34Ch/40Ch	36Ch/42Ch	38Ch/44Ch
EP_AVAIL	310h	330h/3F0h	350h/410h	370h/430h	390h/450h
EP_DATA	314h	334h/3F4h	354h/414h	374h/434h	394h/454h
EP_DATA1	318h	338h/3F8h	358h/418h	378h/438h	398h/458h
EP_VAL	***Reserved***	33Ch	35Ch	37Ch	39Ch
Reserved	3A0h – 3E0h. 3FCh – 400h, 41Ch – 420h, 43Ch – 440h				

Register 16-59. 300h EP_CFG Endpoint Configuration for EP 0 (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
3:0	**Endpoint Number** Selects the Endpoint Number. This field has no effect on EP 0, which always maintains an Endpoint Number of 0.	RO	Yes	0h
6:4	***Reserved***	RsvdZ	Yes	000b
7	**Endpoint Direction** Selects the endpoint direction. The direction is with respect to the USB Host point of view. This bit is automatically changed, according to the direction specified in the Setup packet. 0 = OUT 1 = IN	RW	Yes	0
9:8	**Endpoint Type** Selects the endpoint type. EP 0 is forced to a Control type. 00b = Control 01b = Isochronous 10b = Bulk 11b = Interrupt	RO	Yes	00b
10	**Endpoint Enable** 1 = Enables this endpoint. This bit has no effect on EP 0, which is always enabled.	RO	Yes	1
11	**Byte Packing Enable** 0 = If de-asserted upper Byte Enables are detected on a PCI Express Write to an IN FIFO, the USB transfer is terminated, and the packet is marked with EOP in the FIFO. 1 = If contiguous Byte Enables are detected on a PCI Express Write to an IN FIFO, the partial DWord is saved, until more bytes are received or the USB transfer is explicitly terminated. (Contiguous Byte Enables are defined as the absence of de-asserted Byte Enables, after an asserted Byte Enable is detected.) Field [18:16] (*EP FIFO Byte Count*) has no effect. **Contiguous Byte Enable examples** – 1000b, 1100b, 1110b, 1111b **Non-contiguous Byte Enable examples** – 0111b, 0101b, 0011b, 0001b	RW	Yes	0
15:12	***Reserved***	RsvdZ	Yes	0h

Register 16-59. 300h EP_CFG Endpoint Configuration for EP 0 (USB Controller) *(Cont.)*

<table>
<tr><th>Bit(s)</th><th colspan="3">Description</th><th>Access</th><th>Serial EEPROM and I²C</th><th>Default</th></tr>
<tr><td rowspan="7">18:16</td><td colspan="3">EP FIFO Byte Count
Used to mask the PCI Express Byte Enables for a 1-DWord Write to an EP 0 FIFO, using EP 0's EP_DATA register (USB Controller, offset 314h). A Mask bit value of 1 indicates that the corresponding PCI Express Byte Enable is forced to 0.
Following the next FIFO Write transaction, this field is restored to its default value. If the value is less than 100b, and Byte Packing is disabled, the next EP_DATA register FIFO Write transaction causes a short packet to be validated. This field is programmed to its default value when the corresponding FIFO is flushed.
Encodings not listed are reserved.</td><td rowspan="7">RW</td><td rowspan="7">Yes</td><td rowspan="7">000b</td></tr>
<tr><th>Value</th><th>Mask</th><th>Function</th></tr>
<tr><td>000b</td><td>1111b</td><td>No bytes are written (default); all PCI Express Byte Enables are masked).</td></tr>
<tr><td>001b</td><td>1110b</td><td>Up to 1 byte is written.</td></tr>
<tr><td>010b</td><td>1100b</td><td>Up to 2 bytes are written.</td></tr>
<tr><td>011b</td><td>1000b</td><td>Up to 3 bytes are written.</td></tr>
<tr><td>100b</td><td>0000b</td><td>Up to 4 bytes are written; no PCI Express Byte Enables are masked.</td></tr>
<tr><td>31:19</td><td colspan="3">Reserved</td><td>RsvdZ</td><td>Yes</td><td>0-0h</td></tr>
</table>

Register 16-60. 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h EP_RSP Endpoint Response for EP 0 and GPEP*x* (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
Note: *Writing 1 to bits [7:0] of this register Clears the corresponding register bits. Writing 1 to bits [15:8] of this register Sets the corresponding register bits. When this register is read, bits [7:0] are duplicated on bits [15:8]. If a 1 is simultaneously written to both the Set and Clear bits, the Set bit has priority.*				
0	**Endpoint Halt Clear** Clears the *Endpoint Halt* bit. When an Endpoint Clear Feature Standard Request to the *Endpoint Halt* bit is detected by the CPU, the CPU must write 1 to this bit. Reading this bit returns the current state of the *Endpoint Halt* bit. For EP 0: • *Endpoint Halt* bit is automatically Cleared when another Setup packet is received • *Control Status Stage Handshake* bit should be Cleared when this bit is Set Writing this bit also Clears the *Endpoint Toggle* bit.	RW1C	Yes	0
1	**Endpoint Toggle Clear** Clears the *Endpoint Toggle* bit. The *Endpoint Toggle* bit automatically changes state after each *USB r2.0* or *USB r3.0* packet is successfully transferred. Reading this bit returns the current state of the *Endpoint Toggle* bit. Firmware Clears the *Endpoint Toggle* bit only for Endpoint initialization events.	RW1C	Yes	0
2	**NAK OUT Packets Mode Clear** ***Used only for OUT endpoints.*** Selects the response if the CPU did not process the previously received packet. 0 = Non-Blocking mode. The USB 3382 accepts the OUT packet if there is sufficient space in the endpoint's FIFO. 1 = Blocking mode. The USB 3382 responds to an OUT packet with a NAK/NRDY if the *NAK Packets* bit is Set.	RW1C	Yes	1
3	**Control Status Stage Handshake Clear** ***Used only for EP 0.*** This bit is automatically Set when a Setup packet is detected. After the bit is Cleared, the USB 3382 returns the proper response (NAK/NRDY) to the Host's Control Status stage (ACK for Control Reads and Zero-Length packets for Control Writes). 1 = Control Status stage is acknowledged with a NAK/NRDY	RW1C	Yes	0

Register 16-60. 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h EP_RSP Endpoint Response for EP 0 and GPEP*x* (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
4	**Interrupt Mode Clear** ***Valid only for INTERRUPT endpoints.*** 0 = Standard Interrupt data. Standard Data Toggle protocol is followed. 1 = Isochronous Rate Feedback mode. Interrupt endpoint is used for isochronous rate feedback information. In this mode, the endpoint *Data* toggle bit is changed after each packet is sent to the Host, without regard to handshaking. No packet Retries are performed in this mode.	RW1C	Yes	0
5	**Endpoint Force CRC Error Clear** 1 = All IN packets transmitted, and OUT packets received, by this endpoint are forced to incur a CRC error	RW1C	Yes	0
6	**EP Hide Status Stage Clear** 1 = The following endpoint **EP_STAT** register bits (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch) are not Set for Control Status Stage packets: • *Data Packet Transmitted Interrupt* (bit 2) • *Data Packet Received Interrupt* (bit 3) • *Short OUT Packet Received Interrupt* (bit 5) • *USB OUT ACK Sent* (bit 16) • *USB OUT NAK Sent* (bit 17) • *USB IN ACK Rcvd* (bit 18) • *USB IN NAK Sent* (bit 19) • *USB STALL Sent* (bit 20) • *Timeout* (bit 21) ***Notes:*** *This bit is* ***not*** *used for standard operation, and is intended for use only with special applications.* *The endpoint's* ***EP_STAT*** *register Data OUT/PING Token Interrupt and Data IN Token Interrupt bits (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[1:0], respectively) remain Set, independent of this bit.*	RW1C	Yes	0
7	**NAK Packets Clear** If this bit is Set and another OUT token is received, a NAK/NRDY is returned to the Host if another OUT packet is sent to this endpoint. This bit can be Cleared by a bit in the endpoint's **EP_STAT** register (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch). 1 = The endpoint received a short Data packet from the Host, and the *NAK OUT Packets Mode* bit is Set. For an IN endpoint, all IN tokens cause a NAK/NRDY handshake to be returned to the USB Host.	RW1C	Yes	0

Register 16-60. 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h EP_RSP Endpoint Response for EP 0 and GPEP*x* (USB Controller) ***(Cont.)***

Bit(s)	Description	Access	Serial EEPROM and I^{2}C	Default
8	**Endpoint Halt Set** Refer to Note at beginning of this register.	RW1S	Yes	0
9	**Endpoint Toggle Set** Refer to Note at beginning of this register.	RW1S	Yes	0
10	**NAK OUT Packets Mode Set** Refer to Note at beginning of this register.	RW1S	Yes	1
11	**Control Status Stage Handshake Set** Refer to Note at beginning of this register.	RW1S	Yes	0
12	**Interrupt Mode Set** Refer to Note at beginning of this register.	RW1S	Yes	0
13	**Endpoint Force CRC Error Set** Refer to Note at beginning of this register.	RW1S	Yes	0
14	**EP Hide Status Stage Set** Refer to Note at beginning of this register.	RW1S	Yes	0
15	**NAK Packets Set** Refer to Note at beginning of this register.	RW1S	Yes	0
31:16	***Reserved***	RsvdZ	Yes	0000h

Register 16-61. 308h, 328h/3E8h, 348h/408h, 368h/428h, 388h/448h EP_IRQENB Endpoint Interrupt Enable for EP 0 and GPEP*x* (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**Data IN Token Interrupt Enable** 1 = Enables the endpoint's **EP_STAT** register *Data IN Token Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[0]), which enables an interrupt to be generated when a Data IN token is received from the Host	RW	Yes	0
1	**Data OUT/PING Token Interrupt Enable** 1 = Enables the endpoint's **EP_STAT** register *Data OUT/PING Token Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[0]), which enables an interrupt to be generated when a Data OUT or PING token is received from the Host	RW	Yes	0
2	**Data Packet Transmitted Interrupt Enable** 1 = Enables the endpoint's **EP_STAT** register *Data Packet Transmitted Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[2]), which enables an interrupt to be generated when a Data packet is transmitted to the Host	RW	Yes	0
3	**Data Packet Received Interrupt Enable** 1 = Enables the endpoint's **EP_STAT** register *Data Packet Received Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[3]), which enables an interrupt to be generated when a Data packet is received from the Host	RW	Yes	0
4	***Reserved***	RW	Yes	0
5	**Short OUT Packet Received Interrupt Enable** 1 = Enables the endpoint's **EP_STAT** register *Short OUT Packet Received Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[5]), which enables an interrupt to be generated when the length of the last OUT packet was less than the Maximum Packet Size (determined by the **GPEP[3:0/Out/In]_FS_MAXPKT**, **GPEP[3:0/Out/In]_HS_MAXPKT**, or **GPEP[3:0/Out/In]_SS_MAXPKT** register)	RW	Yes	0
6	**Short OUT Packet Done Interrupt Enable** 1 = Enables the endpoint's **EP_STAT** register *Short OUT Packet Done Interrupt* bit(s) (USB Controller, offset(s) 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[6]), which enables an interrupt to be generated when an OUT FIFO becomes empty after a short (or zero-length) packet is received from the USB Host	RW	Yes	0
12:7	***Reserved***	RsvdZ	Yes	0-0h
13	**ZLP Interrupt Enable** 1 = Enables an interrupt to be generated when the endpoint receives a zero-length packet (ZLP)	RW	Yes	0
14	**DMA Channel Interrupt Enable** 1 = Enables an interrupt to be generated when the endpoint DMA channel interrupt is asserted	RW	Yes	0
31:15	***Reserved***	RsvdZ	Yes	0-0h

Register 16-62. 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch EP_STAT Endpoint Status for EP 0 and GPEP*x* (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
Notes: *When the Auto-Enumerate Controller is servicing EP 0 Standard Requests, this register's Status bits are not Set.* ***Synchronization Considerations*** *– Due to internal synchronization delays, bits [11, 10, 5, and 3] (FIFO Full, FIFO Empty, Short OUT Packet Received Interrupt, and Data Packet Received Interrupt, respectively) can change at slightly different times (up to 100 ns apart). Implement firmware that avoids confusion due to interpreting combinations of bits, particularly bits within the same register.* *For example, if the FIFO is empty and a new OUT packet is received, the Data Packet Received Interrupt bit can be Set up to 600 ns before the FIFO Empty bit returns false and the Short OUT Packet Received Interrupt bit is Set. Firmware reading this register within that window might interpret this as an OUT packet with 0 bytes (because the FIFO Empty bit is not yet Cleared), and not correctly as a short packet (because the Short OUT Packet Received Interrupt status is not yet Set).* *Firmware polling for the above Status bits to change must read the register a second time, after the bit status has changed, to receive consistent information.*				
0	**Data IN Token Interrupt** Writing 1 Clears this bit. 1 = Endpoint received a Data IN token from the Host	RW1C	Yes	0
1	**Data OUT/PING Token Interrupt** Writing 1 Clears this bit. 1 = Endpoint received a Data OUT or PING token from the Host	RW1C	Yes	0
2	**Data Packet Transmitted Interrupt** Writing 1 Clears this bit. 1 = Endpoint transmitted a Data packet to the Host	RW1C	Yes	0
3	**Data Packet Received Interrupt** Writing 1 Clears this bit. 1 = Endpoint received a Data packet from the Host	RW1C	Yes	0

Register 16-62. 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch EP_STAT Endpoint Status for EP 0 and GPEP*x* (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
4	**NAK Packets** This bit can also be controlled by the endpoint's **EP_RSP** register(s) (USB Controller, offset(s) 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h). Writing 1 Clears this status bit. 1 = Endpoint received a short Data packet from the Host, and the **EP_RSP** register *NAK OUT Packets Mode* bit is Set. When another OUT token is received, a NAK/NRDY is returned to the Host when another OUT packet is sent to this endpoint. When Set for an IN endpoint, all IN tokens cause a NAK/NRDY handshake to be returned to the USB Host.	RW1C	Yes	0
5	**Short OUT Packet Received Interrupt** 1 = Length of the last OUT packet received was less than the Maximum Packet Size. Remains Set until a 1 is written to Clear the bit.	RW1C	Yes	0
6	**Short OUT Packet Done Interrupt** To Clear this bit, bit 5 (*Short OUT Packet Received Interrupt*) must also be Cleared (at the same time or earlier). 1 = OUT FIFO became empty after a short (or zero-length) packet was received from the USB Host. Subsequent OUT packets (after the short packet) are prevented from entering the FIFO, by Setting the endpoint's **EP_RSP** register *NAK OUT Packets Mode* bit(s) (USB Controller, offset(s) 304h, 324h/3E4h, 344h/404h, 364h/424h, 384h/444h)	RW1C	Yes	0
7	**Short Packet Transferred Status** 0 = Length of the last packet was the Maximum Packet Size 1 = Length of the last IN or OUT packet was less than the Maximum Packet Size	RO	Yes	0
8	**FIFO Validate** ***Applies only to GPIOx IN endpoints.*** Writing 1 causes invalidated data in the endpoint's FIFO to be validated. Invalidated data can be determined by the Endpoint "Length" Counter. If there is no invalidated data in the FIFO, a Zero-Length packet is written. If the last Write to the endpoint's FIFO had partial Byte Enables asserted, or the **EP_CFG** register *EP FIFO Byte Count* field(s) (USB Controller, offset(s) 320h, 340h, 360h, 380h[18:16]) was not a value of 100b, the next Write to this bit is ignored. If the endpoint's FIFO is full when this bit is written, validation is delayed until space becomes available in the FIFO. This bit is self-Clearing, and reading always returns a value of 0.	RW1S	Yes	0
9	**FIFO Flush** Writing 1 causes the endpoint's FIFO to be flushed, and field [28:24] *(FIFO Valid Counter*) and the endpoint's **EP_AVAIL** register(s) (USB Controller, offset(s) 310h, 330h/3F0h, 350h/410h, 370h/430h, 390h/450h) to be Cleared. This bit is self-Clearing, and reading always returns a value of 0.	RW1S	Yes	0
10	**FIFO Empty** 1 = Endpoint's FIFO is empty	RO	Yes	1
11	**FIFO Full** 1 = Endpoint's FIFO is full	RO	Yes	0

Register 16-62. 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch EP_STAT Endpoint Status for EP 0 and GPEP*x* (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
12	***Reserved***	RsvdZ	Yes	0
13	**ZLP Interrupt** Writing 1 Clears this bit. 1 = Endpoint received a zero-length packet	RW1C	Yes	0
15:14	***Reserved***	RsvdZ	Yes	00b
16	**USB OUT ACK Sent** Writing 1 Clears this bit. 1 = Last USB OUT Data packet transferred was successfully acknowledged with an ACK to the Host	RW1C	Yes	0
17	**USB OUT NAK Sent** Writing 1 Clears this bit. 1 = Last USB OUT Data packet was not accepted, and a NAK/NRDY handshake was returned to the USB Host	RW1C	Yes	0
18	**USB IN ACK Rcvd** Writing 1 Clears this bit. 1 = Last USB IN Data packet transferred was successfully acknowledged with an ACK from the Host	RW1C	Yes	0
19	**USB IN NAK Sent** Writing 1 Clears this bit. 1 = Last USB IN packet was not provided, and a NAK/NRDY handshake was returned to the USB Host	RW1C	Yes	0

Register 16-62. 30Ch, 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch EP_STAT Endpoint Status for EP 0 and GPEP*x* (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
20	**USB STALL Sent** Writing 1 Clears this bit. 1 = Last USB packet was not accepted nor provided, and a STALL handshake was returned to the USB Host	RW1C	Yes	0
21	**Timeout** For an OUT endpoint, if the last USB packet received had a CRC or Bit-Stuffing error, and was not acknowledged by the USB 3382, the USB Host re-transmits the packet. For an IN endpoint, the last USB packet transmitted was not acknowledged by the USB Host, indicating a Bus error. The USB Host expects the same packet to be re-transmitted in response to the next IN token. Writing 1 Clears this bit.	RW1C	Yes	0
23:22	**High-Bandwidth OUT Transaction PID** Provides the PID of the last high-bandwidth OUT packet received. The PID is stable when bit 3 (*Data Packet Received Interrupt*) is Set, and remains stable until another OUT packet is received. 00b = DATA0 01b = DATA1 10b = DATA2 11b = MDATA	RO	Yes	00b
28:24	**FIFO Valid Counter** Provides the quantity of packets currently in the endpoint's IN FIFO. When the value is non-zero, returns a packet in response to IN tokens. The Counter is incremented when a short packet is validated, and decremented when a short packet is successfully sent to the Host. Automatically Cleared by a FIFO Flush operation, –or– when EP 0 receives a Setup packet. The maximum quantity of packets allowed in an IN FIFO is 16.	RO	Yes	0-0h
31:29	***Reserved***	RsvdZ	Yes	000b

Register 16-63. 310h, 330h/3F0h, 350h/410h, 370h/430h, 390h/450h EP_AVAIL Endpoint Available Count for EP 0 and GPEP*x* (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
13:0	**Endpoint Available Counter** **OUT Endpoints** Returns the quantity of valid bytes in the OUT endpoint's FIFO. Values range from 0 (empty) to 4,096 (full). For EP 0, this field provides the FIFO Count only when data is being received from the Host. avail_bytes = [(wrptr_line - rdptr_line - 1) x 8] + (last_valid_byte +1) **IN Endpoints** Returns the quantity of empty bytes in the OUT endpoint's FIFO. Values range from 0 (full) to 4,096 (empty). For EP 0, this field provides the quantity of empty bytes in the FIFO when data is being sent to the Host. avail_space = (fifo_size / 8 - valid_lines) x 8	RO	Yes	0-0h
31:14	***Reserved***	RsvdZ	Yes	0-0h

Register 16-64. 314h, 334h/3F4h, 354h/414h, 374h/434h, 394h/454h EP_DATA Endpoint Data for EP 0 and GPEP*x* (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
31:0	**Endpoint Data** **OUT Endpoints** Used by the CPU to read data from the OUT endpoint's FIFO. **IN Endpoints** Used by the CPU to write data to the endpoint's FIFO. The endpoint's FIFO can also be accessed through **PCI BAR2 and/or PCI BAR3**. Writes to a full FIFO, or for an OUT endpoint, are silently dropped. Reads from an empty FIFO returns a Completion with Data = FFFF_FFFFh if the **FIFOCTL** register *Ignore FIFO Availability* bit (USB Controller, offset 38h[3])is Set, or a Completion with Completer Abort status if the *Ignore FIFO Availability* bit is Cleared. (Refer to Section 8.5.1, "IN FIFO Writes," for further details.) Reads from this register for an IN endpoint return a Completion with Completer Abort status.	RW	Yes	0000_0000h

Register 16-65. 318h, 338h/3F8h, 358h/418h, 378h/438h, 398h/458h EP_DATA1 Endpoint Data1 for EP 0 and GPEP*x* (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
2:0	**Endpoint Data Last Valid Byte** Returns the *Last Valid Byte* field corresponding to the FIFO line where the endpoint's **EP_DATA** register(s) (USB Controller, offset(s) 314h, 334h/3F4h, 354h/414h, 374h/434h, 394h/454h) is being read. This register must be read immediately after reading **EP_DATA**.	RO	Yes	000b
3	**Endpoint Data End of Packet** Returns the *End of Packet* bit corresponding to the FIFO line where the endpoint's **EP_DATA** register(s) is being read. This register must be read immediately after reading **EP_DATA**.	RO	Yes	0
31:4	***Reserved***	RsvdZ	Yes	0000_000h

Register 16-66. 33Ch, 35Ch, 37Ch, 39Ch EP_VAL Endpoint Validate for GPEP*x* (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
31:0	**Endpoint Validate** Writing any value to this register causes a short packet to be validated in the endpoint's IN FIFO. Has the same effect as writing to the endpoint's **EP_STAT** register *FIFO Validate* bit(s) (USB Controller, offset(s) 32Ch/3ECh, 34Ch/40Ch, 36Ch/42Ch, 38Ch/44Ch[8]). Intended for use by an external DMA Controller that uses an extra Descriptor to write to this register, to validate a short packet. Reading this register always returns a value of 0000_0000h.	RW	Yes	0000_0000h

Register 16-67. 320h, 340h, 360h, 380h EP_CFG Endpoint Configuration for GPEP*x* Endpoints (USB Controller)

Bit(s)	Description	Access	Serial EEPROM	Default
In Legacy Adapter mode, each GPEP*x* **EP_CFG** register represents an endpoint that can either be OUT or IN. In Enhanced Adapter mode, each GPEP*x* **EP_CFG** register represents an IN and OUT endpoint pair.				
3:0	**Legacy Adapter Mode** **Endpoint Number** Selects the Endpoint Number. Valid numbers are 1 to Fh.	RW	Yes	0h
	Enhanced Adapter Mode **Endpoint Number** Selects the Endpoint Number. Valid numbers are 1 to Fh.	RW	Yes	Adapter mode: 0h for all Root Complex mode: GPEP0 = 2h GPEP1 = 4h GPEP2 = 6h GPEP3 = 8h
6:4	***Reserved***	RsvdZ	Yes	000b
7	**Legacy Adapter Mode** **Endpoint Direction** Selects the endpoint direction. The direction is with respect to the USB Host point of view. A maximum of one OUT and IN endpoint is allowed for each Endpoint Number. 0 = OUT 1 = IN	RW	Yes	0
	Enhanced Adapter Mode ***Reserved***	RsvdZ	Yes	0

Register 16-67. 320h, 340h, 360h, 380h EP_CFG Endpoint Configuration for GPEP*x* Endpoints (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM	Default
9:8	**Legacy Adapter Mode** **Endpoint Type** Selects the endpoint type. 00b = ***Reserved*** 01b = Isochronous 10b = Bulk 11b = Interrupt	RW	Yes	10b
	Enhanced Adapter Mode **OUT Endpoint Type** Selects the OUT endpoint type. 00b = ***Reserved*** 01b = Isochronous 10b = Bulk 11b = Interrupt	RW	Yes	10b
10	**Legacy Adapter Mode** **Endpoint Enable** 1 = Enables the endpoint	RW	Yes	0
	Enhanced Adapter Mode **OUT Endpoint Enable** 1 = Enables this OUT endpoint	RW	Yes	0 (Adapter mode) 1 (Root Complex mode)
11	**Byte Packing Enable** 0 = If de-asserted upper Byte Enables are detected on a PCI Express Write to an IN FIFO, the USB transfer is terminated, and the packet is marked with EOP in the FIFO. 1 = If contiguous Byte Enables are detected on a PCI Express Write to an IN FIFO, the partial DWord is saved until more bytes are received or the USB transfer is explicitly terminated. (Contiguous Byte Enables are defined as the absence of de-asserted Byte Enables after an asserted Byte Enable is detected.) Field [18:16] (*EP FIFO Byte Count*) has no effect. **Contiguous Byte Enable examples** – 1000b, 1100b, 1110b, 1111b **Non-contiguous Byte Enable examples** – 0111b, 0101b, 0011b, 0001b	RW	Yes	0

Register 16-67. 320h, 340h, 360h, 380h EP_CFG Endpoint Configuration for GPEP*x* Endpoints (USB Controller) *(Cont.)*

Bit(s)	Description	Access	Serial EEPROM	Default
13:12	**Legacy Adapter Mode** ***Reserved***	RsvdZ	Yes	00b
	Enhanced Adapter Mode **IN Endpoint Type** Selects the IN endpoint type. 00b = ***Reserved*** 01b = Isochronous 10b = Bulk 11b = Interrupt	RW	Yes	10b
14	**Legacy Adapter Mode** ***Reserved***	RsvdZ	Yes	0
	Enhanced Adapter Mode **IN Endpoint Enable** 1 = Enables this IN endpoint	RW	Yes	0 (Adapter mode) 1 (Root Complex mode)
15	**IN EP Format** Selects either Stream or Message format. 0 = Stream format is selected. In this mode, only the PCI Express Payload is written to an IN FIFO. 1 = Message format is selected. In this mode, the entire PCI Express packet is written to an IN FIFO.	RW	Yes	0

Register 16-67. 320h, 340h, 360h, 380h EP_CFG Endpoint Configuration for GPEP*x* Endpoints (USB Controller) *(Cont.)*

<table>
<tr><th>Bit(s)</th><th>Description</th><th>Access</th><th>Serial EEPROM</th><th>Default</th></tr>
<tr><td>18:16</td><td>EP FIFO Byte Count
Used to mask the PCI Express Byte Enables for a 1-DWord Write to a GPEPx FIFO, using the endpoint's EP_DATA register(s) (USB Controller, offset(s) 334h/3F4h, 354h/414h, 374h/434h, 394h/454h). A Mask bit value of 1 indicates that the corresponding PCI Express Byte Enable is forced to 0.
Following the next FIFO Write transaction, this field is restored to its default value. If the value is less than 100b, and Byte Packing is disabled, the next EP_DATA register FIFO Write transaction causes a short packet to be validated. This field is programmed to its default value when the corresponding FIFO is flushed.
Encodings not listed are reserved.
<table>
<tr><th>Value</th><th>Mask</th><th>Function</th></tr>
<tr><td>000b</td><td>1111b</td><td>No bytes are written; all PCI Express Byte Enables are masked).</td></tr>
<tr><td>001b</td><td>1110b</td><td>Up to 1 byte is written.</td></tr>
<tr><td>010b</td><td>1100b</td><td>Up to 2 bytes are written.</td></tr>
<tr><td>011b</td><td>1000b</td><td>Up to 3 bytes are written.</td></tr>
<tr><td>100b</td><td>0000b</td><td>Up to 4 bytes are written (default); no PCI Express Byte Enables are masked.</td></tr>
</table></td><td>RW</td><td>Yes</td><td>100b</td></tr>
<tr><td>23:19</td><td>Service Interval
Determines the service interval for Periodic endpoints. Values must be within the range of 1 to 16. The actual interval is $2^{(binterval-1)}$, in units of 125 µs.</td><td>RW</td><td>Yes</td><td>0-0h</td></tr>
<tr><td>27:24</td><td>Max Burst Size
Indicates the quantity of packets that the endpoint can send or receive, as part of a burst. The actual burst size used depends upon the size of the endpoint's FIFO. The total burst length must not be greater than the FIFO size.
Value ranges from 0 to 15, representing burst sizes of 1 to 16.</td><td>RW</td><td>Yes</td><td>0h</td></tr>
<tr><td>29:28</td><td>Sync Type
If ISO, indicates the supported synchronization types.
00b = No synchronization
01b = Asynchronous
10b = Adaptive
11b = Synchronous</td><td>RW</td><td>Yes</td><td>00b</td></tr>
<tr><td>31:30</td><td>Usage Type
If interrupt and ISO, indicates the supported usage types.
<table>
<tr><th>Value</th><th>Interrupt</th><th>ISO</th></tr>
<tr><td>00b</td><td>Periodic</td><td>Data endpoint</td></tr>
<tr><td>01b</td><td>Notification</td><td>Feedback endpoint</td></tr>
<tr><td>10b</td><td>Reserved</td><td>Implicit feedback data endpoint</td></tr>
<tr><td>11b</td><td>Reserved</td><td>Reserved</td></tr>
</table></td><td>RW</td><td>Yes</td><td>00b</td></tr>
</table>

16.11 FIFO Registers

The FIFO registers determine the size and location (Base address) of each USB endpoint FIFO within the USB Ingress and PCI Express Ingress RAM.

Table 16-10. FIFO Registers

Register	Offset (from BAR0)[a]						
	EP 0	GPEP0	GPEP1	GPEP2	GPEP3	PCIOUT/PCIIN	RCIN
EP_FIFO_SIZE_BASE	500h	520h	540h	560h	580h	5E0h	600h

a. *Offsets 504h through 514h, 524h through 534h, 544h through 554h, 564h through 574h, 584h through 594h, 5E4h through 5F4h, and 604h through 614h, are* ***Reserved*** *or* ***Factory Test Only****.*

Register 16-68. 500h, 520h, 540h, 560h, 580h, 5E0h, 600h EP_FIFO_SIZE_BASE EP 0, GPEP0, GPEP1, GPEP2, GPEP3, PCIOUT/PCIIN, and RCIN FIFO Size and Base Address (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I²C	Default
The USB Ingress RAM provides a total of 16-KB RAM for all OUT endpoints that require FIFO space – EP 0, GPEP[3:0], and PCIOUT. The size of each of these six FIFOs is determined by field [2:0] (*OUT FIFO Size*). The size for each FIFO can range from 64 to 4,096 bytes; however, the combined size of the six FIFOs must be less than 16,384 bytes. Each FIFO can be located on any 64-byte RAM boundary. The Base address of each FIFO is determined by field [14:6] (*OUT FIFO Base Address*), and is relative to the start of the USB Ingress RAM. The Base address resolution is 64 bytes. The IN FIFOs – EP 0, GPEP[3:0], PCIIN, and RCIN – are placed into two 4,976-byte RAM segments. The size of each of these seven FIFOs is determined by field [18:16] (*IN FIFO Size*). The size for each FIFO can range from 64 to 4,096 bytes; however, the combined size of the seven FIFOs must be less than 9,952 bytes. Each FIFO can be located on any 64-byte boundary, and must fit entirely within one of the two RAM segments. The Base address of each FIFO is determined by field [30:22] (*IN FIFO Base Address*), and is relative to the start of the PCI Express Ingress RAM. The Base address resolution is 64 bytes.				
2:0	**OUT FIFO Size** Determines the endpoint's OUT FIFO size (in bytes). 000b = 64 001b = 128 010b = 256 011b = 512 100b = 1,024 101b = 2,048 110b = 4,096 111b = ***Reserved***	RW	Yes	Adapter mode – Refer to Table 16-11 Root Complex mode – Refer to Table 16-12
5:3	***Reserved***	RsvdZ	Yes	000b
14:6	**OUT FIFO Base Address** Determines the endpoint's OUT FIFO Base address within the USB Ingress RAM. The address is aligned to a 64-byte boundary; therefore, this field represents Address bits [14:6].	RW	Yes	Adapter mode – Refer to Table 16-11 Root Complex mode – Refer to Table 16-12
15	***Reserved***	RsvdZ	Yes	0
18:16	**IN FIFO Size** Determines the endpoint's IN FIFO size (in bytes). 000b = 64 001b = 128 010b = 256 011b = 512 100b = 1,024 101b = 2,048 110b = 4,096 111b = ***Reserved***	RW	Yes	Adapter mode – Refer to Table 16-11 Root Complex mode – Refer to Table 16-12
21:19	***Reserved***	RsvdZ	Yes	000b
30:22	**IN FIFO Base Address** Determines the endpoint's IN FIFO Base address within the PCI Express Ingress RAM. The address is aligned to a 64-byte boundary; therefore, this field represents Address bits [14:6].	RW	Yes	Adapter mode – Refer to Table 16-11 Root Complex mode – Refer to Table 16-12
31	***Reserved***	RsvdZ	Yes	0

16.11.1 FIFO Size and Base Address Defaults – Adapter Mode

The IN FIFO Base addresses listed in Table 16-11 are relative to the start of the 32-KB PCI Express interface Ingress RAM, in units of 64 bytes. The size value shown in parentheses is the FIFO size, in bytes.

The OUT FIFO Base addresses listed in Table 16-11 are relative to the start of the 32-KB USB interface Ingress RAM, in units of 64 bytes. The base value shown in parentheses is the Base address, in bytes.

Note: *All IN FIFOs must have a gap between them of at least 64 bytes.*

Table 16-11. FIFO Size and Base Address Defaults for Register Offset(s) 500h, 520h, 540h, 560h, 580h, 5E0h, 600h – Adapter Mode

Register	USB Device Endpoint						
	EP 0	GPEP0	GPEP1	GPEP2	GPEP3	PCIOUT/ PCIIN	RCIN
EP_FIFOSIZE (OUT)[a]	3 (512)	6 (4K)	6 (4K)	5 (2K)	5 (2K)	2 (256)	–
EP_FIFOSIZE (IN)[a]	3 (512)	5 (2K)	5 (2K)	5 (2K)	5 (2K)	2 (256)	–
EP_FIFOBASE (OUT)	0	8 (200h)	48h (1200h)	88h (2200h)	A8h (2A00h)	CAh (3280h)	–
EP_FIFOBASE (IN)	0	9 (240h)	2Ah (A80h)	180h (6000h)	1A1h (6840h)	1C4h (7100h)	–

a. Unit of measure is in bytes.

16.11.2 FIFO Size and Base Address Defaults – Root Complex Mode

The IN FIFO Base addresses listed in Table 16-12 are relative to the start of the 32-KB PCI Express interface Ingress RAM, in units of 64 bytes. The size value shown in parentheses is the FIFO size, in bytes.

The OUT FIFO Base addresses listed in Table 16-12 are relative to the start of the 32-KB USB interface Ingress RAM, in units of 64 bytes. The base value shown in parentheses is the Base address, in bytes.

Note: *All IN FIFOs must have a gap between them of at least 64 bytes.*

Table 16-12. FIFO Size and Base Address Defaults for Register Offset(s) 500h, 520h, 540h, 560h, 580h, 5E0h, 600h – Root Complex Mode

Register	USB Device Endpoint						
	EP 0	GPEP0	GPEP1[a]	GPEP2	GPEP3[a]	PCIOUT/ PCIIN	RCIN
EP_FIFOSIZE (OUT)[b]	3 (512)	6 (4K)	2 (256)	6 (4K)	2 (256)	2 (256)	–
EP_FIFOSIZE (IN)[b]	3 (512)	5 (2K)	2 (256)	5 (2K)	2 (256)	2 (256)	3 (512)
EP_FIFOBASE (OUT)	0	8 (200h)	48h (1200h)	4Ch (1300h)	8Ch (2300h)	B4h (2D00h)	–
EP_FIFOBASE (IN)	0	9 (240h)	180h (6000h)	2Ah (A80h)	185h (6140h)	1B0h (6C00h)	1B5h (6D40h)

a. If endpoints GPEP1 or GPEP3 are to be used for traffic that uses USB packets larger than 256 bytes, the default FIFO sizes for these endpoints must be increased.

b. Unit of measure is in bytes.

16.12 USB Power Management Registers

Table 16-13. Power Management Registers

Offset (from BAR0)	Register	Description
6C0h	USBPM Control	USB Power Management Control
6C4h	USBPM Status	USB Power Management Status Register

Register 16-69. 6C0h USBPM Control USB Power Management Control (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**USB Mode PME to ACK Send Enable** 1 = In Adapter mode, enables PME TO ACK Message generation in response to a received PME_Turn_Off Message	RW	Yes	1
1	**PME to ACK in Suspend Only** 1 = In Adapter mode, enables PME to ACK generation only when the USB interface is in the Suspend state	RW	Yes	0
2	**USB2 L1 STATIN Pending Device Remote Wakeup Enable** 1 = *USB r2.0* L1 Link PM state Remote Wakeup due to a pending STATIN Dedicated Endpoint interrupt. The USB 3382 returns a NYET handshake for L1 Requests, if a STATIN interrupt is pending.	RW	Yes	0
3	**USB2 L1 USB IN FIFO Packet Pending Device Remote Wakeup Enable** 1 = Enables *USB r2.0* L1 Link PM state Remote Wakeup due to a pending USB IN FIFO Data packet. The USB 3382 returns a NYET handshake for L1 Requests, if the USB IN FIFOs are not empty.	RW	Yes	0
4	**USB Suspend STATIN Pending Device Remote Wakeup Enable** 1 = Enables USB Suspend Device Remote Wakeup due to a pending STATIN interrupt	RW	Yes	0
5	**USB Suspend IN FIFO Packet Pending Device Remote Wakeup Enable** 1 = Enables USB Suspend Device Remote Wakeup due to a pending USB IN FIFO Data packet	RW	Yes	0
6	***USB r3.0* U2 State Clock Stop Enable** 1 = Enables main clock stop when in the *USB r3.0* U2 Link state	RW	Yes	0
7	**Suspend State Clock Stop Enable** 1 = Enables main clock stop when in the USB Suspend state	RW	Yes	0

Register 16-69. 6C0h USBPM Control USB Power Management Control (USB Controller) ***(Cont.)***

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
8	***USB r3.0* U2 State PLL Stop Enable** 1 = Enables PLL stop when in the *USB r3.0* U2 Link state	RW	Yes	0
9	**Suspend State PLL Stop Enable** 1 = Enables PLL stop when in the Suspend state	RW	Yes	0
10	**USB Suspend Power Turn-Off Enable** Enables switching Off the power when the USB 3382 in the Suspend state. In systems where power in the Vaux state is controlled by the PCI Express Host CPU, this bit should be Cleared. In systems where power to the USB 3382 in Vaux is controlled by the LANE_GOOD0# pin, this bit should be Set.	RW	Yes	0
11	**GPIO0 Software Control** Set this bit if software must allow control of the GPIO0/LANE_GOOD0# signal (*such as* to program I/O or read Lane status). 1 = Allows GPIO0 software control in Bridge mode	RW	Yes	0
12	**PCIe L2L3 USB Disconnect Enable** 1 = In Adapter mode, the connection to the USB Host is removed, by Clearing the **USBCTL** register *USB Detect Enable* bit (USB Controller, offset 8Ch[3]), when Port 0 enters the L2/L3 Ready Link PM state and the USB side is not in the Suspend state	RW	Yes	1
13	**WAKE PIN Suspend Remote Wake Enable** 1 = When in USB suspend, WAKE# Low assertion enables Remote Wakeup	RW	Yes	0
14	**SS Mode UTMI Suspend** 1 = When operating in SuperSpeed mode, enables shutting of the *USB r3.0* Transceiver Macrocell Interface (UTMI), to reduce power consumption	RW	Yes	1
15	**VBUS Debounce Disable**	RW	Yes	0
16	**Bypass Suspend Immediate Timer** 1 = Bypasses the 500 µs Suspend Timer wait time, when going into the Suspend state	RW	Yes	0
17	**USB Unconfigured PCIe PHY Hold Enable** 1 = Disables Port PHY training process from starting, until the USB 3382 is USB-configured. (Configuration value is not 0.)	RW	Yes	0
18	**VBUS Change PCIe Host Wake Enable** 1 = In the D3cold state, enables waking up the PCI Express Host through the WAKE# signal or beacon	RW	Yes	0
20:19	***Not used***	RW	Yes	00b
31:21	***Reserved***	RsvdP	No	0-0h

Register 16-70. 6C4h USBPM Status USB Power Management Status Register (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**Was Vaux Suspend** 1 = Indicates that the USB 3382 came out of the Vaux power-down state, after going into USB suspend	RW1C	Yes	0
2:1	**PCIe Link PM State P0** Port 0 PCI Express Link PM state. 00b = PCI Express Power Management is not active (could be in the L0 Link PM state, or the Link is down) 01b = ASPM L1 Link PM state 10b = PCI Express PM L1 Link PM state 11b = L2/L3 Ready Link PM state	RO	No	00b
4:3	**PCIe Link PM State P1** Port 0 PCI Express Link PM state. 00b = PCI Express Power Management is not active (could be in the L0 Link PM state, or the Link is down) 01b = ASPM L1 Link PM state 10b = PCI Express PM L1 Link PM state 11b = L2/L3 Ready Link PM state	RO	No	00b
6:5	**USB Link PM State** 00b = USB Power Management is not active (could be in the U0 Link state, or the Link is down) 01b = U1 Link state 10b = U2 Link state/L1 Link PM state 11b = U3 Link state/L3 Link PM state	RO	No	00b
31:7	***Reserved***	RsvdP	No	0-0h

16.13 Indexed Registers

Indexed registers are accessed by way of the USB Controller Memory-Mapped **IDXADDR** and **IDXDATA** registers (USB Controller, offsets 30h and 34h, respectively). *For example*, to access the **DIAG** register (Index 00h), software must first write the index value, 00h, to the **IDXADDR** register. Reads from the **IDXDATA** register then return the value for the Indexed register. Writes to the **IDXDATA** register update the indexed register.

Table 16-14. Indexed Registers

Index	Register	Description
Index 00h	DIAG	Diagnostic Control
Index 01h	PKTLEN	Packet Length
Index 02h	FRAME	Frame Counter
Index 03h	CHIPREV	Chip Revision
Index 04h	UFRAME	Micro Frame Counter
Index 05h	FRAME_COUNT	Frame Down Counter
Index 06h	HS_MAXPOWER	High-Speed Maximum Power
Index 07h	FS_MAXPOWER	Full-Speed Maximum Power
Index 08h	HS_INTPOLL_RATE	High-Speed Interrupt Polling Rate
Index 09h	FS_INTPOLL_RATE	Full-Speed Interrupt Polling Rate
Index 0Ah	HS_NAK_RATE	High-Speed NAK Rate
Index 0Bh	SCRATCH	Scratchpad
Index 0Ch – 1Fh	***Reserved***	
Index 20h + (*n* x 10h) Index 20h, 30h, 40h, 50h, 60h, 70h, 80h, 90h	GPEP[3:0/Out/In]_HS_MAXPKT	High-Speed Maximum Packet Size
21h + (*n* x 10h) Index 21h, 31h, 41h, 51h, 61h, 71h, 81h, 91h	GPEP[3:0/Out/In]_FS_MAXPKT	Full-Speed Maximum Packet Size
22h + (*n* x 10h) Index 22h, 32h, 42h, 52h, 62h, 72h, 82h, 92h	GPEP[3:0/Out/In]_SS_MAXPKT	SuperSpeed Maximum Packet Size
Index 23h – 2Fh Index 33h – 3Fh Index 43h – 4Fh Index 53h – 5Fh Index 63h – 6Fh Index 73h – 7Fh Index 83h	***Reserved***	
Index 84h	STATIN_HS_INTPOLL_RATE	High-Speed Interrupt Polling Rate for STATIN
Index 85h	STATIN_FS_INTPOLL_RATE	Full-Speed Interrupt Polling Rate for STATIN
Index 86h	SS_MAXPOWER	SuperSpeed Maximum Power
Index 87h - 8Ch Index 93h - FFh	***Reserved***	

Register 16-71. Index 00h DIAG Diagnostic Control (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
0	**Force Transmit CRC Error** 1 = CRC error is forced on the next transmitted Data packet. Inverting the msb of the calculated CRC generates the CRC error. Automatically Cleared at the end of the next packet.	RW1S	Yes	0
1	***Reserved***	RsvdZ	Yes	0
2	**Force Receive Error** 1 = Error is forced on the next received Data packet. As a result, the packet is not acknowledged. Automatically Cleared at the end of the next packet.	RW1S	Yes	0
3	***Reserved***	RsvdZ	Yes	0
4	**Fast Times** 1 = Internal timers and Counters operate at a fast speed for ***Factory Test Only***	RW	Yes	0
5	**Illegal Byte Enables** Writing 1 Clears this bit. 1 = External PCI Express agent performed a transaction to and/or from one of the endpoint FIFOs, and illegal First Byte Enables were detected. The only valid Byte Enable combinations are 0001b, 0011b, 0111b, and 1111b.	RW1C	Yes	0
7:6	***Reserved***	RsvdZ	Yes	00b
8	**Force CPU Interrupt** 1 = Forces an interrupt to the 8051	RW	Yes	0
9	**Force USB Interrupt** 1 = Forces a STATIN endpoint interrupt to generate	RW	Yes	0
10	**Force PCI Express Interrupt** 1 = In Adapter mode, forces an MSI to be sent to the PCI Express Root Complex	RW	Yes	0
11	***Reserved***	RW	Yes	0
12	**USB RAM 1-Bit Soft Error Injection** 1 = Every toggle injects a 1-bit error	RW	Yes	0
13	**USB RAM 2-Bit Soft Error Injection** 1 = Every toggle injects a 2-bit error	RW	Yes	0
14	**USB RAM Error Injection Field Select** 0 = Error injection is in the *ECC Code* field 1 = Error injection is in the *Data* field	RW	Yes	0
15	***Reserved***	RsvdZ	Yes	0
31:16	**Retry Counter** Accumulator of packet Retries. Cleared when written with a value.	RW1C	Yes	0000h

Register 16-72. Index 01h PKTLEN Packet Length (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
10:0	**RX Packet Length** Provides the last packet length received. This field is not updated when Setup packets are received, because they maintain a fixed length of 8. This field is intended for debugging purposes, and might be in the process of being updated by the USB Controller when being read. Read twice to verify the value.	RO	Yes	0-0h
15:11	***Reserved***	RsvdZ	Yes	0-0h
26:16	**TX Packet Length** Provides the last packet length transmitted. This field is intended for debugging purposes, and might be in the process of being updated by the USB section when being read. Read twice to verify the value.	RO	Yes	0-0h
31:27	***Reserved***	RsvdZ	Yes	0-0h

Register 16-73. Index 02h FRAME Frame Counter (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
10:0	**Frame Counter** Contains the Frame Counter from the most recent Start-of-Frame (SOF) packet.	RO	Yes	0-0h
31:11	***Reserved***	RsvdZ	Yes	0-0h

Register 16-74. Index 03h CHIPREV Chip Revision (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
15:0	**Chip Revision** Returns the current USB 3382 Silicon Revision.	RO	Yes	Silicon Revision AA: 00AAh Silicon Revision AB: 00ABh
31:16	***Reserved***	RsvdZ	Yes	0000h

Register 16-75. Index 04h UFRAME Micro Frame Counter (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
2:0	**Micro Frame Counter** ***Valid only in High-Speed mode.*** Contains the Micro Frame Counter from the most recent Micro Start-of-Frame (SOF) packet.	RO	Yes	000b
4:3	***Reserved***	RsvdZ	Yes	00b
18:5	**ITP Bus Interval Counter** ***Valid only in SuperSpeed mode.*** Contains the *Bus Interval Counter* field from the most recent ITP.	RO	Yes	0-0h
31:19	**ITP Delta** ***Valid only in SuperSpeed mode.*** Contains the *Delta* field from the most recent ITP.	RO	Yes	0-0h

Register 16-76. Index 05h FRAME_COUNT Frame Down Counter (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
10:0	**Frame Re-Load** Contains the value that is automatically loaded into the Frame Downcount Counter when the Counter reaches a value of 0 and a new SOF or ITP is detected.	RW	Yes	0-0h
15:11	***Reserved***	RsvdZ	Yes	0-0h
26:16	**Frame Downcount** Contains a Down Counter that is decremented when an SOF or ITP is detected on the USB. When this Counter reflects a value of 0 and a new SOF or ITP is detected, the Counter is loaded with the value in field [10:0] (*Frame Re-Load*), and an interrupt is generated (**IRQSTAT1** register *SOF Downcount Interrupt Status* bit (USB Controller, offset 2Ch[14]) is Set).	RW	Yes	0-0h
31:27	***Reserved***	RsvdZ	Yes	0-0h

Register 16-77. Index 06h HS_MAXPOWER High-Speed Maximum Power (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
7:0	**High-Speed Maximum Power** ***Used only when the Get Configuration Descriptor Request is in Auto-Enumerate mode, and the USB 3382 is operating in High-Speed mode.*** The amount of current drawn by the peripheral from the USB Controller, in increments of 2 mA. The USB 3382 reports this value to the USB Host, in the Configuration Descriptor. The default for a two-PCI Express Lane configuration is 160 mA (50h x 2 mA).	RW	Yes	0 PCI Express Lanes = 3Ch 1 PCI Express Lane = 44h 2 PCI Express Lanes = 50h
31:8	***Reserved***	RsvdZ	Yes	0000_00h

Register 16-78. Index 07h FS_MAXPOWER Full-Speed Maximum Power (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
7:0	**Full-Speed Maximum Power** ***Used only when the Get Configuration Descriptor Request is in Auto-Enumerate mode, and the USB 3382 is operating in Full-Speed mode.*** The amount of current drawn by the peripheral from the USB Controller, in increments of 2 mA. The USB 3382 reports this value to the USB Host, in the Configuration Descriptor. The default for a two-PCI Express Lane configuration is 160 mA (50h x 2 mA).	RW	Yes	0 PCI Express Lanes – 3Ch 1 PCI Express Lane – 44h 2 PCI Express Lanes – 50h
31:8	***Reserved***	RsvdZ	Yes	0000_00h

Register 16-79. Index 08h HS_INTPOLL_RATE High-Speed Interrupt Polling Rate (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
7:0	**High-Speed Interrupt Polling Rate** Specifies the interrupt polling rate, in terms of microframes (125 µs). Returned as the last byte of all Interrupt Endpoint Descriptors, when the Get Configuration Descriptor is Set to Auto-Enumerate mode, and the USB 3382 is operating in High-Speed mode.	RW	Yes	FFh
31:8	***Reserved***	RsvdZ	Yes	0000_00h

Register 16-80. Index 09h FS_INTPOLL_RATE Full-Speed Interrupt Polling Rate (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
7:0	**Full-Speed Interrupt Polling Rate** Specifies the interrupt polling rate, in milliseconds. Returned as the last byte of all Interrupt Endpoint Descriptors, when the Get Configuration Descriptor is Set to Auto-Enumerate mode, and the USB 3382 is operating in Full-Speed mode.	RW	Yes	FFh
31:8	***Reserved***	RsvdZ	Yes	0000_00h

Register 16-81. Index 0Ah HS_NAK_RATE High-Speed NAK Rate (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
7:0	**High-Speed NAK Rate** Specifies the maximum NAK rate of High-Speed Bulk/Control endpoints, in response to OUT packets. A value of 0 indicates the endpoint never NAKs an OUT packet. Other values indicate, at most, 1 NAK each HS_NAK_RATE quantity of microframes. Value ranges from 0 to 255.	RW	Yes	00h
31:8	***Reserved***	RsvdZ	Yes	0000_00h

Register 16-82. Index 0Bh SCRATCH Scratchpad (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
31:0	**Scratchpad** General-purpose Scratchpad register.	RW	Yes	FEED_FACEh

16.13.1 Maximum Packet Size – GPEP*x* Index Registers

Even-numbered GPEPs are OUT endpoints, and odd-numbered GPEPs are IN endpoints. In Legacy mode, the values in the GPEP OUT registers are used for both the IN and OUT directions of a GPEP.

Table 16-15. GPEP*x* Index Register Summary

GPEP*x*	High-Speed Maximum Packet Index	Full-Speed Maximum Packet Index	SuperSpeed Maximum Packet Index
0 OUT	20h	21h	22h
1 OUT	30h	31h	32h
2 OUT	40h	41h	42h
3 OUT	50h	51h	52h
0 IN	60h	61h	62h
1 IN	70h	71h	72h
2 IN	80h	81h	82h
3 IN	90h	91h	92h

Register 16-83. Index 20h, 30h, 40h, 50h, 60h, 70h, 80h, 90h GPEP[3:0/Out/In]_HS_MAXPKT High-Speed Maximum Packet Size (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
10:0	**High-Speed Maximum Packet Size** Determines the endpoint Maximum Packet Size, when operating in High-Speed mode. For Interrupt and Isochronous endpoints, the Maximum Packet Size must be a multiple of 8 bytes.	RW	Yes	200h
12:11	**Additional Transaction Opportunities** Determines the quantity of additional transaction opportunities, per microframe, for High-Speed Isochronous and Interrupt endpoints. 00b = None (one transaction per microframe) 01b = One additional transaction opportunity (two per microframe) 10b = Two additional transaction opportunities (three per microframe) 11b = ***Reserved***	RW	Yes	00b
31:13	***Reserved***	RsvdZ	Yes	0-0h

Register 16-84. Index 21h, 31h, 41h, 51h, 61h, 71h, 81h, 91h GPEP[3:0/Out/In]_FS_MAXPKT Full-Speed Maximum Packet Size (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
10:0	**Full-Speed Maximum Packet Size** Determines the endpoint Maximum Packet Size, when operating in Full-Speed mode. For Interrupt and Isochronous endpoints, the Maximum Packet Size must be a multiple of 8 bytes.	RW	Yes	40h
31:11	***Reserved***	RsvdZ	Yes	0-0h

Register 16-85. Index 22h, 32h, 42h, 52h, 62h, 72h, 82h, 92h GPEP[3:0/Out/In]_SS_MAXPKT SuperSpeed Maximum Packet Size (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
10:0	**SuperSpeed Maximum Packet Size** Determines the endpoint Maximum Packet Size, when operating in SuperSpeed mode. For Interrupt and Isochronous endpoints, the Maximum Packet Size must be a multiple of 8 bytes.	RW	Yes	400h
12:11	**Max Packets per Service Interval** Determines the maximum quantity of packets within a service interval that this endpoint supports. Maximum quantity of packets = *bMaxBurst* x (this field + 1) The maximum value for this field is 10b.	RW	Yes	00b
15:13	***Reserved***	RsvdZ	Yes	000b
31:16	**Total Bytes per Service Interval** ***Valid only for Periodic endpoints.*** The total quantity of bytes that this endpoint transfers every service interval.	RW	Yes	0000h

Register 16-86. Index 84h STATIN_HS_INTPOLL_RATE High-Speed Interrupt Polling Rate for STATIN (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
7:0	**STATIN High-Speed Interrupt Polling Rate** Specifies the interrupt polling rate, in terms of microframes (125 µs). Returned as the last byte of the STATIN Interrupt Endpoint Descriptor, when the Get Configuration Descriptor is Set to Auto-Enumerate mode, and the USB 3382 is operating in High-Speed mode.	RW	Yes	01h
31:8	***Reserved***	RsvdZ	Yes	0000_00h

Register 16-87. Index 85h STATIN_FS_INTPOLL_RATE Full-Speed Interrupt Polling Rate for STATIN (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
7:0	**STATIN Full-Speed Interrupt Polling Rate** Specifies the interrupt polling rate, in milliseconds. Returned as the last byte of the STATIN Interrupt Endpoint Descriptor, when the Get Configuration Descriptor is Set to Auto-Enumerate mode, and the USB 3382 is operating in Full-Speed mode.	RW	Yes	01h
31:8	***Reserved***	RsvdZ	Yes	0000_00h

Register 16-88. Index 86h SS_MAXPOWER SuperSpeed Maximum Power (USB Controller)

Bit(s)	Description	Access	Serial EEPROM and I^2C	Default
7:0	**SuperSpeed Maximum Power** Used only when the Get Configuration Descriptor Request is in Auto-Enumerate mode and the USB 3382 is operating in SuperSpeed mode. Amount of current drawn by the peripheral from the USB Port, in increments of 8 mA. The USB 3382 reports this value to the USB Host in the Configuration Descriptor. The default for a PCI Express two-Lane configuration is 160 mA (14h x 8 mA).	RW	Yes	0 PCI Express Lanes – 0Fh 1 PCI Express Lanes – 11h 2 PCI Express Lanes – 14h
31:8	***Reserved***	RsvdZ	Yes	0000_00h

16.13.2 Maximum Packet Size – Non-GPEP*x* Endpoints

Table 16-16 lists the Maximum Packet Sizes for non-GPEP*x* endpoints.

Table 16-16. Maximum Packet Size (MPS) Constant Values for Non-GPEP*x* Endpoints[a]

Endpoint	Full-Speed MPS	High-Speed MPS	SuperSpeed MPS	MaxPkts/ Interval	TotalBytes/ Interval	Additional Transaction Opportunities
EP 0	64	64	512	0	0	0
CSROUT	64	512	1,024	0	0	0
CSRIN	64	512	1,024	0	0	0
PCIOUT	64	512	1,024	0	0	0
PCIIN	64	512	1,024	0	0	0
STATIN (BULK)	64	512	1,024	0	8	0
STATIN (INTERRUPT)	8	8	8	0	8	0
RCIN (BULK)	64	512	1,024	0	16	0
RCIN (INTERRUPT)	16	16	16	0	16	0

a. *The values listed here are decimal; however, the registers return hex values.*

Chapter 17 Test and Debug

17.1 Introduction

This chapter describes the following test- and debug-related information:

- Physical Layer Loopback Operation
- User Test Pattern
- Pseudo-Random Bit Sequence
- Using the SerDes Diagnostic Data Register
- PHY Testability Features
- JTAG Interface
- Lane Good Status LEDs

17.2 Physical Layer Loopback Operation

17.2.1 Overview

Physical Layer (PHY) Loopback functions are used to test the SerDes in the USB 3382, connections between devices, and SerDes of external devices, as well as various USB 3382 and external digital logic. The USB 3382 supports three types of Loopback operations, as described in Table 17-1. Additional information regarding each type is provided in the sections that follow.

Table 17-1. Loopback Operations

Operation	Description
Analog Loopback Master Mode	This mode depends upon an external device or passive connection (*such as* a cable) to loopback the transmitted data to the USB 3382, without SKIP Ordered-Set clock compensation. If an external device is used, it must not include its Elastic buffer in the Slave Loopback data path, so that SKIP Ordered-Sets are not inserted. A device's re-transmitted Receive data must be sent back to the Master, synchronous to the Master's Transmit Reference Clock. *That is*, the Slave device re-serializes the Transmit data, using the recovered clock from the received data. In that mode, the PRBS generator and checker should be used to create and check the data pattern.
Digital Loopback Master Mode	This mode depends upon an external device to loopback the transmitted data that includes at least its Elastic buffer in the Loopback data path, allowing for reliable loopback testing, in case the two devices have asynchronous Reference Clock sources with Parts per Million (PPM) offsets. The Master's pattern generator inserts SKIP Ordered-Sets at regular intervals, and its received data checker can handle PPM offset clock compensation, by way of SKIP symbol addition or deletion. The USB 3382 provides a User Test Pattern generator and checker that can be used for Digital loopback testing.
Digital Loopback Slave Mode	The USB 3382 enters this mode when an external device transmits Training Sets with the *Loopback* Training Control Bit Set and the **SKIP Ordered-Set Interval** register *Analog Loopback Enable* bit (Port 0, offset 234h[6]) is Cleared. This is the default Loopback mode for the LTSSM Slave *Loopback.Active* substate. In this mode, the data is looped back at the 8-bit level, which includes the USB 3382's Elastic buffer, 8b/10b decoder, and 8b/10b encoder in the Slave Loopback data path. Asynchronous clock compensation can occur in the Elastic buffer through SKIP symbol addition or deletion, depending upon clock PPM offsets and fill threshold decoding. The Master data pattern checker must be able to handle the presence of SKIP Ordered-Sets and variations in their contents, when SKIP Ordered-Sets are transmitted.

17.2.2 Analog Loopback Master Mode

Analog Loopback Master mode is typically used for Analog Far-End testing (refer to Figure 17-1), with a shallow Loopback path Slave device, to determine overall Bit Error rates. However, it can also be used for passive external serial loopback with a cable. Looping back with a cable includes the internal circuitry, package connections to bond pads, package pins, board traces, and any connectors that might be in the test data path, as illustrated in Figure 17-2. A PRBS pattern is typically used for this mode, because it is appropriate for bit error rate testing. A User Test Pattern (UTP) is ***not*** recommended for this application – refer to Section 17.3 for details.

Figure 17-1. Analog Far-End Loopback

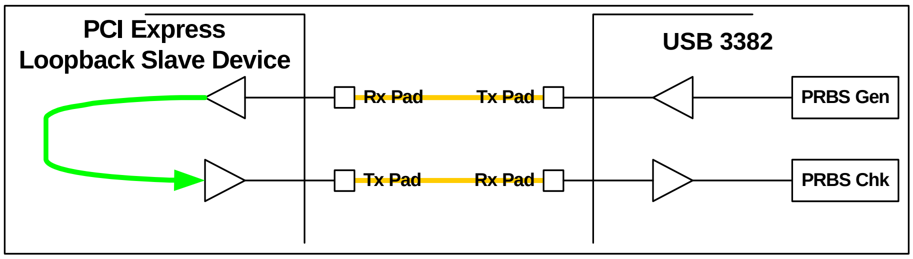
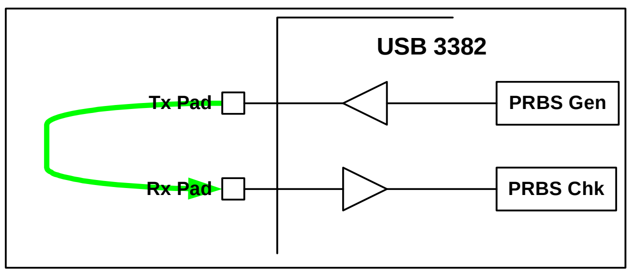


Figure 17-2. Cable Loopback



17.2.2.1 Initiating Far-End Analog Operations in USB 3382 Master Devices

Note: *Initiating a Master Loopback operation on Port 0 can cause a Deadlock condition to occur, unless an I²C Slave interface is used to write and read Configuration Space register bits instead of writing them through Port 0 Configuration transactions. Therefore, it is recommended to restrict Analog Master loopback testing to downstream Ports when external devices are used.*

One way to test Master Analog loopback with passive cables is to have Port 0 connected to a Root Complex, for Configuration Read/Write transactions that are used to Set and monitor the key device register bits. In that case, only downstream Ports would be test-capable, to avoid potential Deadlock conditions on Port 0. Alternatively, an I²C Slave interface and Rapid Development Kit (RDK) software could be used to write or read the registers. This makes any Port testable. The user has the option of attaching one or more cables to the appropriate high-speed Tx and Rx differential pairs that belong to the Ports being tested.

Loopback cables can be attached before or after a standard power-up initialization sequence. If the cables are attached before power-up, use a serial EEPROM to program the Port/SuperSpeed USB's **Physical Layer Port/SuperSpeed USB Command** register *Port x/SuperSpeed USB Loopback Command* bit (Port 0, offset 230h[12, 8, 4, 0]). The bit arms the Port/SuperSpeed USB to enter the Master *Loopback.Entry* substate. When written from a serial EEPROM, the bit's assertion is present before the Ports/SuperSpeed USB begin Link training. In that case, the Ports/SuperSpeed USB directly transition to the LTSSM *Loopback* state from the LTSSM *Configuration* state. The LTSSM exits the *Polling* state and enters the *Configuration.LinkWidth.Start* substate, then immediately transitions to the Master *Loopback.Entry* substate.

At this point, users can sample the Port/SuperSpeed USB's **Physical Layer Port/SuperSpeed USB Command** register *Port x/SuperSpeed USB Ready as Loopback Master* bit (Port 0, offset 230h[15, 11, 7, 3]), to determine whether the bit is Set, which indicates that the Master has reached the LTSSM *Loopback.Active* substate. At this time, the PRBS engine can be enabled, by writing the **SerDes Test** register *SerDes x/SuperSpeed USB BIST Generator/Checker Enable* (PRBS Enable) bit(s) (Port 0, offset B88h[19:16]), for the SerDes/SuperSpeed USB associated with the Port/SuperSpeed USB being tested, with the sequences listed in Table 17-2.

Table 17-2. Sequence to Enable PRBS Transmission(SerDes Test Register (Port 0, offset B88h)

Data	Description
0000_3074h	Select 8/16-bit PRBS
000E_3034h	Disable comma pattern generator, enable error checker
000E_3035h	Reset error checker
000E_3034h	Clear reset

The PRBS Receive data checker first synchronizes the de-serialized parallel data words from the returned pattern with a reference PRBS pattern generator. Once synchronized, the PRBS checker looks for errors, on a continuous basis. Any errors detected are logged in one or more of the **SerDes Diagnostic Data** register RO bits (Port 0, offset 238h[30 and 23:0]). The errors can be retrieved, by reading the appropriate bit.

If the *Port x/SuperSpeed USB Loopback Command* bits are not Set through the serial EEPROM, the Ports'/SuperSpeed USB's Loopback Training Sets can be used to cause the Ports/SuperSpeed USB to linkup, by way of a Configuration cross-link track, resulting with the Ports/SuperSpeed USB being in the L0 Link PM state. This linkup of a Port/SuperSpeed USB, in response to its own Training Sets, works only if the Port/SuperSpeed USB's **Physical Layer Additional Status** register *Port x/ SuperSpeed USB External Loopback Enable* bit (Port 0, offset 254h[19:16]) is Set by serial EEPROM. After the Port/SuperSpeed USB is in the L0 Link PM state, Configuration Space register programming can then be performed manually, to invoke a Master Loopback operation.

After the Ports/SuperSpeed USB linkup, users can direct the Ports/SuperSpeed USB linkup into an Analog Loopback Master condition, by writing the **Physical Layer Port/SuperSpeed USB Command** register *Port x/SuperSpeed USB Loopback Command* bit(s), (Port 0, offset 230h[12, 8, 4, 0]), through Port 0 and/or the I^2C Slave interface. However, this is not sufficient to initiate the LTSSM transition from the L0 Link PM state, to the *Loopback* state. The Link must pass through a *Recovery* substate, before the *Port x/SuperSpeed USB Loopback Command* bits can be sampled and allow the LTSSM to pass through the *Recovery* state to the *Loopback* state. To cause the Port/SuperSpeed USB to enter the *Recovery* state, users must Set the Port/SuperSpeed USB's **Link Control** register *Retrain Link* bit (Downstream Ports, offset 78h[5]). At this point, users should monitor the Port/SuperSpeed USB's *Port x/SuperSpeed USB Ready as Loopback Master* bit(s), and when Set, the PRBS engine(s) can be enabled, as previously described.

If loopback cables are attached after the device powers up, those Ports whose Lanes are floating unconnected did not detect Receivers. Therefore, those Ports are not trained up to the L0 Link PM state.

If the Port/SuperSpeed USB's *Port x/SuperSpeed USB Loopback Command* and *External Loopback Enable* bits for the downstream Ports/SuperSpeed USB to be tested are written ***before*** the cables are attached, then once cabled, there is Receiver detection, the Port(s)/SuperSpeed USB go through Link training, and then exit the LTSSM *Configuration* state and directly enter the *Loopback* state.

However, if the Port/SuperSpeed USB's *Port x/SuperSpeed USB Loopback Command* and *External Loopback Enable* bits are Set ***after*** the cables are attached, the Ports/SuperSpeed USB do not recognize their own Training Sets and will likely cycle back and forth between *Configuration* and *Detect*. Therefore, users must at least Set the *Port x/SuperSpeed USB External Loopback Enable* bit for the Port/ SuperSpeed USB being tested, by way of serial EEPROM, if the USB 3382 is powered up before the cables are attached. Users can then program the Port/SuperSpeed USB's *Port x/SuperSpeed USB Loopback Command* bit(s). In addition to this, a forced retrain is also needed, to enter into the *Loopback* state through the *Recovery* state, as previously described.

17.2.3 Digital Loopback Master Mode

The only difference between Analog and Digital Loopback Master modes is that the external device is assumed to have at least an Elastic buffer in the Loopback data path. Because of this, SKIP Ordered-Sets must be included in the test data pattern, which precludes use of the PRBS engine.

Figure 17-3 illustrates a Far-End Digital Loopback Master connection and data path.

The USB 3382 provides a User Test Pattern engine on a per-Lane basis, for Digital Far-End Loopback testing. The user pattern itself, however, is common to all Lanes where it is enabled. Details on the use of the User Test Pattern registers and controls are described later in Section 17.5.

What is important to note about the data path (not shown in Figure 17-3) is that the pattern generators and checkers in the USB 3382 Digital Loopback Master have 8b/10b encode, 10b/8b decode, and Elastic buffers included in the Tx/Rx path. The scramblers and de-scramblers are disabled. Therefore, the Digital Loopback Slave device must not scramble the returning data. The 10-bit data can be decoded to 8-bit, and encoded back to 10-bit as an option, and will not affect the UTP pattern checker in the USB 3382, unless there is a coding error.

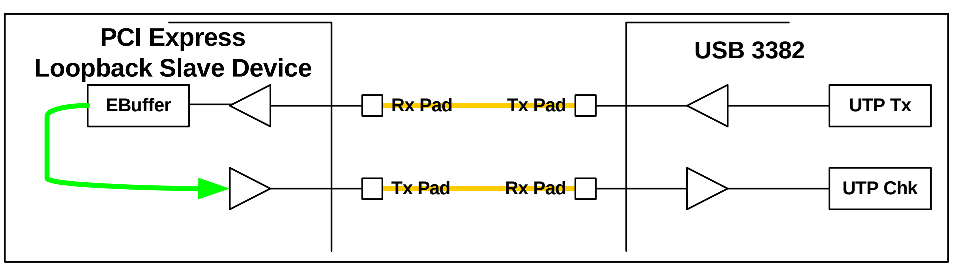
Digital Loopback Master mode is established by either programming method previously described in Section 17.2.2 for Analog Loopback Master mode. The Port/SuperSpeed USB's **Physical Layer Port/ SuperSpeed USB Command** register *Port x/SuperSpeed USB Loopback Command* bit (Port 0, offset 230h[12, 8, 4, 0]) can be Set with a serial EEPROM, causing Loopback to be entered directly from the LTSSM *Configuration* state. Otherwise, the Port/SuperSpeed USB's *Port x/SuperSpeed USB Loopback Command* bit can be Set after linkup, and then the Port/SuperSpeed USB's **Link Control** register *Retrain Link* bit (Downstream Ports, offset 78h[5]) can be used to move the Port/SuperSpeed USB to the *Loopback* state, through the LTSSM *Recovery* state.

After Digital Loopback Master mode is established, Configuration Space register Writes are used to establish a User Test Pattern transmission, as well as error checking, which are described later in Section 17.3.

The UTP is multiplexed, unconditionally, onto the Transmit data path, upon Setting one or more of the **Physical Layer Test** register *SerDes x/SuperSpeed USB User Test Pattern Enable* bit(s) (Port 0, offset 228h[31:28]).

Note: *It is important to verify that the LTSSM is in a Master Loopback.Active substate, before writing 1 to the SerDes x/SuperSpeed USB User Test Pattern Enable bits. Therefore, do not use the serial EEPROM to Set the SerDes x/SuperSpeed USB User Test Pattern Enable bits. (Refer to Section 17.4 for details.)*

Figure 17-3. Digital Far-End Loopback



17.2.4 Digital Loopback Slave Mode

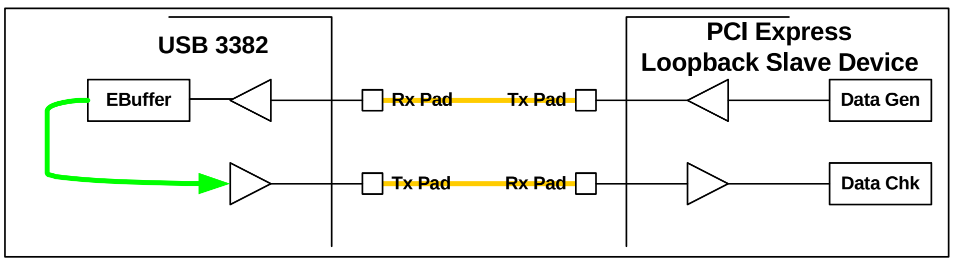
When a Port/the SuperSpeed USB is in the LTSSM Slave *Loopback.Active* substate, it automatically becomes a Digital Loopback Slave, by default. The Port/SuperSpeed USB enters this state after it receives Training Sets with the *Loopback* Training Control Bit Set.

When a Port/the SuperSpeed USB is a Digital Loopback Slave, it includes the Elastic buffer, 8b/10b decoder, and 8b/10b encoder in the Slave Loopback data path. The Loopback Master must provide the test data pattern and data pattern checker (*such as* a USB 3382 User Test Pattern). The Loopback Master must also transmit SKIP Ordered-Sets with the data pattern. Depending upon the USB 3382 Reference Clock source's PPM offset, the USB 3382 Digital Loopback Slave's Elastic buffers can compensate for the offset, by returning more or fewer SKIP symbols than the USB 3382 received from the Master. Therefore, the Master's data pattern checker must make provisions for this when decoding for errors.

This mode is ***not*** suitable for a PRBS pattern as transmitted from the Master, because neither device can compensate for Reference Clock offset differences, should they exist.

To force the Loopback path into Digital Loopback Slave mode, the Slave must be brought into the mode by a Master-connected device, through standard LTSSM tracks.

Figure 17-4. Digital Loopback Slave Mode



17.3 User Test Pattern

The USB 3382 provides a User Test Pattern (UTP) Transmit and Receive data checker, for Digital Far-End Loopback testing. (Refer to Figure 17-3.) After LTSSM Loopback Master mode is established, Configuration Writes are used to fill the Test Data Pattern registers.One or more **Physical Layer Test** register *SerDes x/SuperSpeed USB User Test Pattern Enable* bit(s) (Port 0, offset 228h[31:28]) are used to start the UTP transmission, on the Lanes assigned to each bit. SKIP Ordered-Sets are inserted into the user's test data pattern, at the nearest data pattern boundary according to the programmed SKIP interval. That interval is determined by the **SKIP Ordered-Set Interval** register *SKIP Ordered-Set Interval* field (Port 0, offset 234h[11:0]). The default interval is 1,180 symbol times.

The Test Pattern checker ignores SKIP Ordered-Sets returned by the Loopback Slave, because the quantity of SKIP symbols received can be different from the quantity transmitted. All other data is compared to the transmitted data, and errors are logged in the **SerDes Diagnostic Data** register (Port 0, offset 238h).

The 16-byte UTP is loaded into the **Physical Layer User Test Pattern, Bytes *x* through *y*** registers (Port 0, offsets 210h through 21Ch). The pattern is common to all Lanes. Prior to transmission, the 8b/10b encoder converts the 16 bytes to 10-bit encoded data. Pattern bytes only go out as control symbols (k-bit set), if their corresponding **Physical Layer Command and Status** register *User Test Pattern K-Code Flag* bit (Port 0, offset 220h[31:16]) is Set.

Notice: ***Use care when Setting User Test Pattern K-Code Flag bits, because UTP logic does not check the validity of Control characters.***

The UTP Transmitter logic does not immediately transmit the UTP bytes upon being enabled – a fixed, 8-byte sync pattern (314D_5243h) is transmitted first. The sync word detection validates the physical Loopback wiring and connected device Loopback path, to qualify the UTP transmission's initiation. The sync DWord allows the Pattern Checking logic to determine the starting boundary of the received pattern byte sequence. Sync detection also enables Received Data error checking and logging. There are no sync-acquired status bits in the Physical Layer (PHY) registers; however, the UTP Error Counter saturates at 255d, if the UTP checker does not receive the synchronization word.

Notes: *The **SerDes Diagnostic Data** register UTP/PRBS Error Counter field (Port 0, offset 238h[23:16]) is the UTP Error Counter, when the register's PRBS Counter/-UTP Counter bit (bit 30) is Cleared.*

There are no explicit Control bits for deliberately injecting UTP errors into the transmission, to test the error checking ability. However, one way of testing the ability is to write a test pattern byte to a different value after the transmission has started. That usually causes a temporary unequal boundary condition, which will log an error. While not guaranteed to inject an error, this method is useful for testing error checking ability.

A UTP is not recommended for Master mode far-end cable testing, especially when initiated by way of serial EEPROM from a power-up sequence. If a UTP is enabled and looped back before Link training begins, the symbol framers will not have seen any COM symbols, and the true 10-bit symbol boundaries are unknown. The framer requires three COMs in a row, in the same bit position, to achieve symbol lock. Neither the sync pattern, nor the user pattern, would be detected in this case, and the test is certain to fail.

In addition to the 16-byte pattern registers, the UTP is enabled on a per-Lane basis, by Setting one or more of the **Physical Layer Test** register *SerDes x/SuperSpeed USB User Test Pattern Enable* bit(s) (Port 0, offset 228h[31:28]), for the SerDes/SuperSpeed USB associated with the Port/SuperSpeed USB being tested.

Note: *The UTP is unconditionally multiplexed onto the Transmit data path, upon setting the SerDes x/SuperSpeed USB User Test Pattern Enable bits. Therefore, it is necessary to verify that the LTSSM is in an LTSSM Master Loopback.Active substate* ***before*** *writing those Enable bits to a value of 1.* ***Do not use a serial EEPROM to Set the SerDes x/SuperSpeed USB User Test Pattern Enable bits.***

UTP testing results can be monitored in the **SerDes Diagnostic Data** register (Port 0, offset 238h). The register can be used to examine the Error Count in the *UTP/PRBS Error Counter* field [23:16], and expected/actual data of the first failing byte (fields [7:0 and 15:8], respectively). The *Status* bits are on a per-Lane/SuperSpeed USB basis. **The important field in this register is field [25:24] (*SerDes Diagnostic Data Select*).** When the Lane/SuperSpeed USB code is written to that field, the UTP status for that Lane/SuperSpeed USB appears in the *Status* fields. Bit 30 of the register (*PRBS Counter/-UTP Counter*) is a pattern type indicator, and is Cleared when UTP is enabled for a Lane/SuperSpeed USB.

Notes: *Any errors detected are logged in the* ***SerDes Diagnostic Data*** *register (Port 0, offset 238h). Use of this register is explained in Section 17.5.*

The UTP and PRBS Enables (refer to Section 17.4) are mutually exclusive, and must not be concurrently Set. *If both Enables are concurrently Set, the resulting operation is undefined. The UTP/PRBS Error Counter field continues to count, until it saturates at 255d. To Clear the Counter and allow it to begin logging errors again, write all zeros (0) to that field. Alternatively, the Counter status is Cleared if the SerDes x/SuperSpeed USB User Test Pattern Enable bit for that Lane/SuperSpeed USB is Cleared, and then Set again.*

17.4 Pseudo-Random Bit Sequence

A Pseudo-Random Bit Sequence (PRBS) generator and checker are useful as a diagnostic/debugging tool, and for measuring short- or long-term bit error rates in PCI Express systems. The USB 3382 also uses a specially enabled power-up self-test that runs after reset, as a wafer sort test for use on automated test equipment. PRBS pattern generators and checkers reside within the SerDes_rclk_blk modules, because they transmit and receive 10- or 20-bit data directly to/from the SerDes modules. Locating them in the modules helps ensure tight timing and short trace length on SerDes Tx and Rx parallel data.

The USB 3382 PRBS engine can be enabled, by writing the **SerDes Test** register *SerDes x/SuperSpeed USB BIST Generator/Checker Enable* (PRBS Enable) bit(s) (Port 0, offset B88h[19:16]), for the SerDes/SuperSpeed USB associated with the Port/SuperSpeed USB being tested, with the sequences listed in Table 17-2. Prior to enabling PRBS, an externally connected PCI Express device must be in an LTSSM Slave *Loopback.Active* substate. Furthermore, the reference clocking between the two devices must be synchronous. (*That is*, the returning PRBS pattern must have its transmission clock source synchronous to the USB 3382 Reference Clock.) The USB 3382 PRBS pattern generator does not insert any SKIP Ordered-Sets, and, if the Slave device inserts SKIP Ordered-Sets into the returning pattern, they cannot be ignored by the PRBS checker (it causes an error). Alternatively, the PRBS pattern can be used to test an external cable Loopback, after the correct LTSSM Master *Loopback.Active* substate is reached, as described in Section 17.2.2.

After a Lane/SuperSpeed USB's PRBS engine is enabled, the PRBS engine immediately begins to transmit the PRBS pattern on that Lane/SuperSpeed USB. No 8b/10b encoding is performed. The PRBS pattern generator produces 10- or 20-bit symbols on every Clock cycle, depending upon the current Link speed. The symbols are written directly into the SerDes Tx data Port, for immediate transmission.

The PRBS Receive Data Checking logic first synchronizes the de-serialized 10- or 20-bit Parallel Data symbols from the SerDes Rx data Port, using a reference PRBS pattern generator. After pattern synchronization is achieved, the Receive data checker begins comparing the Rx data symbols on a continuous basis, to discover any mismatch between a symbol's expected and received values.

PRBS testing results can be monitored in the **SerDes Diagnostic Data** register (Port 0, offset 238h). The register can be used to examine the Error Count in the *UTP/PRBS Error Counter* field [23:16]. Expected and actual data is not available when PRBS is used. The *Status* bits are on a per-Lane/ SuperSpeed USB basis. **The important field in this register is field [25:24] (*SerDes Diagnostic Data Select*).** When the Lane/SuperSpeed USB code is written to that field, the PRBS status for that Lane/ SuperSpeed USB appears in the *Status* fields. Bit 30 of the register (*PRBS Counter/-UTP Counter*) is a pattern type indicator, and is Set when PRBS is enabled for a Lane/SuperSpeed USB.

Note: *Any errors detected are logged in the* ***SerDes Diagnostic Data*** *register (Port 0, offset 238h). Use of this register is explained in Section 17.5.*

The PRBS and UTP (refer to Section 17.3) Enables are mutually exclusive, and must not be concurrently Set. *If both Enables are concurrently Set, the resulting operation is undefined. The UTP/PRBS Error Counter field continues to count, until it saturates at 255d. To Clear the Counter and allow it to begin logging errors again, write all zeros (0) to that field. Alternatively, the Counter status is Cleared if the SerDes x/SuperSpeed USB User Test Pattern Enable bit for that Lane/SuperSpeed USB is Cleared, and then Set again.*

The PRBS Error Count does not necessarily represent a true Bit Error rate. The PRBS checker detects one or more mismatched bits in each examined symbol, on a symbol-per-core-clock basis. Therefore, the Error Counter advances one count for every symbol mismatch, regardless of how many bits are in error for that failing symbol.

17.5 Using the SerDes Diagnostic Data Register

The **SerDes Diagnostic Data** register (Port 0, offset 238h) contents reflect the performance of the SerDes/SuperSpeed USB selected by the register's *SerDes Diagnostic Data Select* field [25:24], as defined in Table 17-3. This control is specific to this register, which reports results of UTP and PRBS tests.

Table 17-3. SerDes Register Contents (Port 0, offset 238h)

SerDes Diagnostic Data Select **Field [25:24] Value**			
00b	**01b**	**10b**	**11b**
SerDes 0	SerDes 1	SerDes 2	SuperSpeed USB

17.6 PHY Testability Features

The USB 3382 includes several Configuration bits to ease PHY testability. Features include:

- Full support of the standard and modified compliance patterns
- Register controllability of the common block and Lane-specific inputs of the SerDes

Table 17-4 describes the Configuration bits.

Table 17-4. Configuration Bits to Ease PHY Testability

Register Bit(s)	Description
SerDes x/SuperSpeed USB Mask Electrical Idle Detect **Physical Layer Electrical Idle Detect Mask** register (Port 0, offset 204h[3:0])	**Never Detect Electrical Idle Mask.** When any one of these bits is Set, the Lane/SuperSpeed USB's *Electrical Idle* condition flag does not assert, regardless of the actual presence of Electrical Idle.
SerDes x/SuperSpeed USB Mask Receiver Not Detected **Physical Layer Receiver Not Detected Mask** register (Port 0, offset 204h[19:16])	**Always Detect a Receiver Mask.** When any one of these bits is Set, the PHY functions as if the Lane/SuperSpeed USB detected a Receiver, regardless of the actual presence of a Receiver.
Test Pattern x **Physical Layer User Test Pattern, Bytes *x* through *y*** registers (Port 0, offsets 210h through 21Ch)	A 16-byte test pattern can be written to these four registers. When UTP transmission is enabled, Byte 0 of register offset 210h is transmitted first and Byte 3 (Byte 15 of the UTP) of register offset 21Ch is transmitted last. (Refer to Section 17.2.3 for further details.) Every byte of the UTP can be a Control or Data character. Illegal Control characters can be specified.
Port x/SuperSpeed USB Scrambler Disable Command **Physical Layer Port/SuperSpeed USB Command** register (Port 0, offset 230h[13, 9, 1])	Unconditionally disables the data scramblers on the Lanes of the corresponding Port/SuperSpeed USB, and causes the *Scrambler Disable* Training Control Bit to be Set in transmitted Training Sets. There is one bit for each Port/SuperSpeed USB.
Disable Port x/SuperSpeed USB **Port and SuperSpeed USB Control** register (Port 0, offset 234h[19:16])	When Set, unconditionally disables the Port/SuperSpeed USB. This is different from the LTSSM *Disabled* state, in that the Port/SuperSpeed USB does not attempt to enter this state. If the Port/SuperSpeed USB is idle, it ceases attempting to detect a Receiver. If the Port/SuperSpeed USB is up, it immediately returns to the *Detect.Quiet* substate and remains there. No Electrical Idle Ordered-Set (EIOS) is sent, which could force any connected device to the *Recovery* state, and then to the LTSSM *Detect* state. The Port/SuperSpeed USB remains disabled until its *Disable Port x/SuperSpeed USB* bit is Cleared. While the Port/SuperSpeed USB is disabled, the SerDes that belong to the disabled Port/SuperSpeed USB are placed into the P1 SerDes Power state.
Port x/SuperSpeed USB Quiet **Port and SuperSpeed USB Control** register (Port 0, offset 234h[23:20])	When Set, the LTSSM remains in the *Detect.Quiet* substate on the Port/SuperSpeed USB if it is currently in, or returns to, that substate. Once in the *Detect.Quiet* substate, Receiver termination is enabled and the Transmitters are placed into the P0 SerDes Power state. The Port/SuperSpeed USB can now transmit test patterns (PRBS or UTP), with or without an attached device and without being in the *Loopback.Active* substate.
Port x/SuperSpeed USB Test Pattern x Rate **Port and SuperSpeed USB Control** register (Port 0, offset 234h[27:24])	The Port/SuperSpeed USB transmits the selected test pattern (PRBS or UTP) at 5.0 GT/s, if the Port/SuperSpeed USB's *Port x/SuperSpeed USB* bit is also Set (manual rate selection is enabled only when the *Port x/SuperSpeed USB* bit is Set).
Port x/SuperSpeed USB Receiver Error Counter **Port Receiver Error Counters** register (Port 0, offset 248h)	Contains four 8-bit fields that, when read, return the quantity of Receiver errors detected by the corresponding Port. The Error Counter saturates at 255. The Counter is Cleared with any Write to the corresponding byte in this register; otherwise, this register is RO.

17.7 JTAG Interface

The USB 3382 provides a Joint Test Action Group (JTAG) Boundary Scan interface, which is used to debug board connectivity for each pin.

17.7.1 *IEEE 1149.1* and *IEEE 1149.6* Test Access Port

The *IEEE Standard 1149.1* Test Access Port (TAP), commonly called the *JTAG Debug Port*, is an architectural standard described in the *IEEE Standard 1149.1-1990*. The *IEEE Standard 1149.6-2003* defines extensions to *1149.1* to support PCI Express SerDes testing. These standards describe methods for accessing internal device facilities, using a four- or five-signal interface.

The JTAG Debug Port, originally designed to support scan-based board testing, is enhanced to support the attachment of debug tools. The enhancements, which comply with the *IEEE Standard 1149.1-1994 Specifications for Vendor-Specific Extensions*, are compatible with standard JTAG hardware for boundary-scan system testing.

- **JTAG Signals –** JTAG Debug Port implements the four required JTAG signals – JTAG_TCK, JTAG_TDI, JTAG_TDO, JTAG_TMS – and optional JTAG_TRST# signal
- **Clock Requirements** – JTAG_TCK signal frequency ranges from 0 to 20 MHz
- **JTAG Reset Requirements –** Refer to Section 17.7.4

17.7.2 JTAG Instructions

The JTAG Debug Port provides the *IEEE Standard 1149.1-1990* BYPASS, EXTEST, SAMPLE, PRELOAD, CLAMP, and IDCODE instructions. *IEEE Standard 1149.6-2003* EXTEST_PULSE and EXTEST_TRAIN instructions are also supported. Table 17-5 lists the JTAG instructions, along with their input codes.

The USB 3382 returns the JTAG IDCODE values listed in Table 17-6.

Table 17-5. JTAG Instructions

Instruction	Input Code	Comments
BYPASS	Fh	*IEEE Standard 1149.1-1990*
EXTEST	2h	
SAMPLE	3h	
PRELOAD	3h	
EXTEST_PULSE	9h	*IEEE Standard 1149.6-2003*
EXTEST_TRAIN	8h	
CLAMP	6h	*IEEE Standard 1149.1-1990*
IDCODE	Dh	

Table 17-6. JTAG IDCODE Values

Silicon Revision	Units	Version	Part Number	PLX Manufacturer Identity	Least Significant Bit
AA	Bits	0000b	0011_0011_1000_0010b	001_1100_1101b	1
	Hex	0h	3382h	1CDh	1h
	Decimal	0	13186	461	1
AB	Bits	0001b	0011_0011_1000_0010b	001_1100_1101b	1
	Hex	1h	3382h	1CDh	1h
	Decimal	1	13186	461	1

17.7.3 JTAG Boundary Scan

Boundary Scan Description Language (BSDL), IEEE Standard 1149.1-1994, is a supplement to the *IEEE Standard 1149.1-1990* and *IEEE Standard 1149.1a-1993*, *IEEE Standard Test Access Port and Boundary-Scan Architecture*. BSDL, a subset of the *IEEE 1076-1993 Standard VHSIC Hardware Description Language (VHDL)*, allows a rigorous description of testability features in components which comply with the standard. This standard is used by automated test pattern generation tools for package interconnect tests, and Electronic Design Automation (EDA) tools for synthesized Test logic and verification. BSDL supports robust extensions that can be used for internal test generation and to write software for hardware debug and diagnostics.

The primary components of BSDL include the logical Port description, physical pin map, instruction set, and **Boundary** register description.

The logical Port description assigns symbolic names to the device's signal pins. Each pin includes a logical type of *in*, *out*, *in out*, *buffer*, or *linkage* that defines the logical direction of signal flow.

The physical pin map correlates the device's logical Ports to the physical pins of a specific package. A BSDL description can include several physical pin maps, and maps are provided with a unique name.

Instruction Set statements describe the bit patterns that must be shifted into the **Instruction** register to place the device in the various test modes defined by the standard. Instruction Set statements also support descriptions of instructions that are unique to the USB 3382.

The **Boundary** register description lists each cell or shift stage of the **Boundary** register. Each cell has a unique number, the cell numbered 0 is the closest to the Test Data Out (JTAG_TDO) pin and the cell with the highest number is closest to the Test Data In (JTAG_TDI) pin. Each cell includes additional information, *such as*:

- Cell type
- Logical Port associated with the cell
- Logical function of the cell
- Safe value
- Control cell number
- Disable value
- Result value

17.7.4 JTAG Reset Input – JTAG_TRST#

The JTAG_TRST# input is the asynchronous JTAG logic reset. When JTAG_TRST# is Low, it causes the USB 3382's JTAG TAP Controller to initialize. In addition, when the JTAG TAP Controller is initialized, it selects the USB 3382 standard logic path (core-to-I/O). It is recommended to take the following into consideration when implementing the asynchronous JTAG logic reset on a board:

- If JTAG functionality is required, consider one of the following:
  - JTAG_TRST# Input signal to use a Low-to-High transition once during USB 3382 boot-up, along with the system PEX_PERST# signal
  - Hold the JTAG_TMS pin High while clocking the JTAG_TCK pin five times
- If JTAG functionality is not required, the JTAG_TRST# signal must be directly connected to VSS (Ground), to hold the JTAG TAP Controller inactive
- If the USB 3382's JTAG TAP Controller is not intended to be used by the design, it is recommended that a 1.5KΩ pull-down resistor be connected to the JTAG_TRST# pin, to hold the JTAG TAP Controller in the *Test-Logic-Reset* state, which enables standard logic operation

17.8 Lane Good Status LEDs

The USB 3382 provides Lane Good outputs, LANE_GOOD[1:0]#, that can be used to control external circuitry, *such as* LEDs, to provide visual indication that the PHY of each Lane/SuperSpeed USB's Link is trained to at least x1 width.

Note: *For bridges, the default functionality of the LANE_GOOD[1:0]# pins is GPIO[1:0].*

Software can determine:

- Which Lanes/SuperSpeed USB have completed PHY linkup, by performing a Memory Read of the **Software Lane Status** register *Lane x/SuperSpeed USB Up Status* bits (Port 0, offset 1F4h[3:0], which correspond to Lanes [3-0], respectively).
- Whether the Port/SuperSpeed USB's Link has trained, by reading the **VC0 Resource Status** register *VC0 Negotiation Pending* bit (All Ports and USB Controller, offset 160h[17]) in each Port/SuperSpeed USB. If the *VC0 Negotiation Pending* bit is Cleared, the Link has completed Flow Control (FC) initialization.

  The **VC0 Resource Status** register can be read by either a PCI Express Enhanced Configuration access or Memory Read.
- The negotiated Link width of Ports 0 and 1, by reading the **Link Status** register *Negotiated Link Width* field (All Ports and USB Controller, offset 78h[25:20]) of Ports 0 and 1. This register can be read by either a Configuration Request or Memory Read.

Table 17-7 describes the relationship of the LED On/Off patterns, as they relate to the Lane status indicated by LANE_GOOD[1:0]#.

Table 17-7. LANE_GOOD[1:0]# LED On/Off Patterns, by State

State	LED Pattern
Lane is disabled	Solid Off
Lane is enabled, 5.0 GT/s	Solid On
Lane is enabled, 2.5 GT/s	0.5 seconds On, 0.5 seconds Off

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 18 Electrical Specifications

18.1 Introduction

This chapter provides the USB 3382 electrical specifications.

18.2 Power-Up/Power-Down Sequence

The USB 3382 does not have power-sequencing requirements. The power rails can be powered up and powered down, in any sequence.

18.3 Absolute Maximum Ratings

Notice: ***Maximum limits indicate the temperatures and voltages above which permanent damage can occur. Proper operation at these conditions is not guaranteed, and continuous operation of the USB 3382 at these limits is not recommended.***

Table 18-1. Absolute Maximum Rating (All Voltages Referenced to VSS System Ground)

Item	Symbol	Absolute Maximum Rating	Units
SerDes Analog Supply Voltage	PEX_VDDA_P2 PEX_VDDA_P1 PEX_VDDA_P0 USB_VDDA	-0.5 to +4.6	V
SerDes Digital Supply Voltage	PEX_VDDD0_P2 PEX_VDDD0_P1 PEX_VDDD0_P0 USB_VDDD0 PEX_VDDD1_P2 PEX_VDDD1_P1 PEX_VDDD1_P0 USB_VDDD1	-0.5 to +1.4	V
PLL Supply Voltage	PLL_AVDD	-0.5 to +1.4	V
Auxiliary Core (Logic) Supply Voltage	VAUX_CORE	-0.5 to +1.4	V
Auxiliary I/O (Logic) Supply Voltage	VAUX_IO	-0.5 to +4.6	V
Core (Logic) Supply Voltage	VDD_CORE	-0.5 to +1.4	V
I/O Interface Supply Voltage, 3.3V	VDD_IO USB_AVDD33 USB_VDD33	-0.5 to +4.6	V
Input Voltage (3.3V Interface)	V_I	-0.5 to +4.6	V
Operating Ambient Temperature (Industrial)	T_A	-40 to +85	°C
Storage Temperature	T_{STG}	-65 to +125	°C

18.4 Power Characteristics

Table 18-2. Operating Condition Power Supply Rails

Symbol	Parameter	Min	Typ	Max	Units
PEX_VDDA_P2 PEX_VDDA_P1 PEX_VDDA_P0 USB_VDDA	Analog SerDes Supply[a]	3.00	3.30	3.60	V
PEX_VDDD0_P2 PEX_VDDD0_P1 PEX_VDDD0_P0 USB_VDDD0 PEX_VDDD1_P2 PEX_VDDD1_P1 PEX_VDDD1_P0 USB_VDDD1	Digital SerDes Supply	0.95	1.00	1.10	V
PLL_AVDD	Analog PLL Supply	0.95	1.00	1.10	V
VAUX_CORE	Auxiliary Digital Core Supply	0.95	1.00	1.10	V
VAUX_IO	Auxiliary I/O Supply	3.00	3.30	3.60	V
VDD_CORE	Digital Core Supply	0.95	1.00	1.10	V
VDD_IO USB_AVDD33 USB_VDD33	I/O Supply	3.00	3.30	3.60	V

a. Must be the same voltage as VDD_IO.

18.5 Power Consumption Estimates

Table 18-3. Power Consumption Estimates

Lanes[e]	Ports[e]	Core Logic[a]		SerDes Analog[b]		SerDes Digital[c]		I/O[d]		Total	
		Typ	Max	Typ	Max	Typ	Max	Typ	Max	Typ[f]	Max[g]
		(milliWatts)									
3	3	234	374	99	139	327	468	128	174	788	1,154
		(milliAmps)									
		234	374	30	42	327	468	39	53	–	–

a. *Core Logic supply consists of VDD_CORE.*

b. *SerDes Analog supply consists of PEX_VDDA_P2, PEX_VDDA_P1, and PEX_VDDA_P0.*

c. *SerDes Digital supply consists of PEX_VDDD0_P2, PEX_VDDD0_P1, PEX_VDDD0_P0, PEX_VDDD1_P2, PEX_VDDD1_P1, PEX_VDDD1_P0, PLL_AVDD, USB_VDDD1, and USB_VDDD0.*

d. *I/O supply consists of VDD_IO and USB_VDD33.*

e. *"Lanes" consist of two Lanes of PCI Express (5.0 GT/s), plus one SuperSpeed USB Link (5 Gbps). The two PCI Express Lanes can be combined as single x2 Gen 1 Port, or they can be configured as two x1 Gen 1 or Gen 2 Ports (Ports 0 and 1). The external Ports are part of a three-Port PCI Express switch, the third Port of which (internal Port 2) connects to the USB Controller (PCI Express Adapter (endpoint)), and central RAM. When Port 0 is configured as an upstream Port, the USB 3382 can exist as a device in a USB Controller endpoint. When Port 0 is configured as a downstream Port, the USB 3382 functions as a PCI Express Root Complex, sitting at the top of a PCI Express hierarchy.*

f. *Typical power based upon all Lanes active (L0 Link PM state), 100% traffic, typical voltages (1.00V, 3.30V), and room temperature (25°C).*

g. *Maximum power based upon all Lanes active (L0 Link PM state), maximum traffic, maximum voltages (1.10V, 3.60V), maximum temperature (85°C) and Fast-Fast (FF) process corner silicon.*

18.6 I/O Interface Signal Groupings

Table 18-4. Signal Group PCI Express Analog Interface

Signal Group	Signal Type	Signals	Notes
(a)	PCI Express Output (Transmit)	PEX_PETn[1:0], PEX_PETp[1:0]	Refer to Table 18-7 and Table 18-8
(b)	PCI Express Input (Receive)	PEX_PERn[1:0], PEX_PERp[1:0]	Refer to Table 18-7 and Table 18-9
(c)	PCI Express Differential Clock Input	PEX_REFCLKn, PEX_REFCLKp	Refer to Table 18-7 and Table 18-10

Table 18-5. Signal Group USB Analog Interface

Signal Group	Signal Type	Signals	Notes
(d)	SuperSpeed *USB r3.0* Output (Transmit)	USB_TXM, USB_TXP	Refer to Table 18-7 and Table 18-8
(e)	SuperSpeed *USB r3.0* Input (Receive)	USB_RXM, USB_RXP	Refer to Table 18-7 and Table 18-9
(f)	*USB r2.0* Input/Output	USB_DM, USB_DP	Refer to Table 18-7 and Table 18-10
(g)	External Reference Resistor	USB_RREF	1.6KΩ ±1%, and refer to Table 18-7

Table 18-6. Signal Group Digital Interface

Signal Group	Signal Type	Signals	Note
(h)	Digital Input	CPU_RXD, USB_VBUS	Refer to Table 18-7
(i)	Digital Output	CPU_TXD, EE_WRDATA/EE_DI, FATAL_ERR#, JTAG_TDO, PROCMON	Refer to Table 18-7
(j)	Digital Input with Internal 50KΩ Pull-Up Resistor[a]	I2C_ADDR[2:0], JTAG_TCK, JTAG_TDI, JTAG_TMS, JTAG_TRST#, PEX_PERST#, PWRON_RST#, STRAP_DEBUG_SEL#, STRAP_FAST_BRINGUP#, STRAP_LEGACY, STRAP_PLL_BYPASS#, STRAP_PROBE_MODE#, STRAP_SERDES_MODE_EN#, STRAP_SMBUS_EN#, STRAP_SSC_CENTER#, STRAP_TESTMODE[3, 1], STRAP_UPCFG_TIMER_EN#	Refer to Table 18-7
(k)	Digital Input with Internal 50KΩ Pull-Down Resistor	STRAP_PORTCFG, STRAP_RC_MODE, STRAP_TESTMODE[2, 0]	Refer to Table 18-7
(l)	Bidirectional I/O Buffer, 3.3V, with Internal 50KΩ Pull-Up Resistor	EE_CS#, EE_RDDATA/EE_DO, EE_CLK/ EE_SK, GPIO[3:2], LANE_GOOD[1:0]#	Refer to Table 18-7
(m)	Bidirectional (Open Drain) with Internal 50KΩ Pull-Up Resistor	PEX_INTA#, WAKE#	Refer to Table 18-7
(n)	Bidirectional (Open Drain) I/O Buffer, 3.3V, with Schmitt-Trigger Input	I2C_SCL, I2C_SDA	Refer to Table 18-7
(o)	Manufacturing Test Input with Internal Pull-down Resistor	MFG_AMC, MFG_TAPEN, MFG_TMC1, MFG_TMC2	Refer to Table 18-7
(p)	External Crystal Oscillator Input	XTAL_IN	Refer to Table 18-7
(q)	External Crystal Oscillator Output	XTAL_OUT	Refer to Table 18-7

a. *These signals must be pulled High to VDD_IO or Low to Ground, per the instructions provided in Section 2.4, "Signal Pin Descriptions."*

Table 18-7. Analog and Digital Interfaces (All Signal Groups) – DC Electrical Characteristics

Symbol	Signal Group(s)	Parameter	Min	Typ	Max	Unit	Conditions
I_{OL}	i, l, n	Output Low Current at 3.3V	6.0	11.1	14.9	mA	V_{OL} = 0.4V
I_{OH}	i, l, n	Output High Current at 3.3V	6.0	17.3	28.2	mA	V_{OH} = 2.4V
V_{IL}	h, l	Input Low Voltage at 3.3V	-0.3		0.8	V	
V_{IH}	h, l	Input High Voltage at 3.3V	2.0		3.6	V	
V_N	n	Schmitt Input at 3.3V	0.6		1.1	V	
V_P	n	Schmitt Input at 3.3V	1.2		2.1	V	
V_H	n	Schmitt Input at 3.3V	0.3		1.5	V	
C_{PIN}		Ball Capacitance			5	pF	
$I_{LEAKAGE}$		Input Leakage			±10	µA	
R_{PU}	l	Pull-Up Impedance	33.6K	50K	69.3K	Ω	
R_{PD}	l	Pull-Down Impedance	33.5K	50K	69.4K	Ω	

Table 18-8. 2.5 and 5.0 GT/s PCI Express Transmitter (Signal Group a) – AC and DC Characteristics

Symbol	Parameter	2.5 GT/s	5.0 GT/s	Units	Comments
UI	Unit Interval	399.88 (min) 400.12 (max)	199.94 (min) 200.06 (max)	ps	The specified UI is equivalent to a tolerance of ±300 ppm. UI does not account for variations caused by Spread-Spectrum Clock (SSC). Refer to Note 1.
$V_{TX-DIFF-PP}$	Differential Peak-to-Peak Output Voltage	0.8 (min) 1.2 (max)	0.8 (min) 1.2 (max)	V	Measured with compliance test load. $V_{TX-DIFF-PP} = 2 \times \lvert V_{TX-D+} - V_{TX-D-} \rvert$
$V_{TX-DIFF-PP-LOW}$	Low Power Differential Peak-to-Peak Output Voltage	0.4 (min) 1.2 (max)	0.4 (min) 1.2 (max)	V	Measured with compliance test load. $V_{TX-DIFF-PP-LOW} = 2 \times \lvert V_{TX-D+} - V_{TX-D-} \rvert$ Must be implemented with no de-emphasis.
$V_{TX-DE-RATIO-3.5dB}$	Tx De-Emphasis Level Ratio	3.0 (min) 4.0 (max)	3.0 (min) 4.0 (max)	dB	Ratio of the $V_{TX-DIFF-PP}$ of the 2^{nd} and following bits after a transition, divided by the $V_{TX-DIFF-PP}$ of the 1^{st} bit after a transition. Refer to Note 2.
$V_{TX-DE-RATIO-6dB}$	Tx De-Emphasis Level Ratio	N/A	5.5 (min) 6.5 (max)	dB	Ratio of the $V_{TX-DIFF-PP}$ of the 2^{nd} and following bits after a transition, divided by the $V_{TX-DIFF-PP}$ of the 1^{st} bit after a transition. Refer to Note 2.
$T_{MIN-PULSE}$	Instantaneous Pulse Width (including all jitter sources)	Not specified	0.9 (min)	UI	Measured relative to rising/falling pulse. Refer to Note 3.
T_{TX-EYE}	Minimum Tx Eye Width	0.75 (min)	0.75 (min)	UI	Does not include SSC nor REFCLK jitter. Includes Rj at 10^{-12}. Refer to Notes 3 and 4.
$T_{TX-EYE-MEDIAN-to-MAX-JITTER}$	Maximum Time between the Jitter Median and Maximum Deviation from the Median	0.125 (max)	Not specified	UI	Measured differentially at zero crossing points, after applying the 2.5 GT/s Clock Recovery function. Refer to Note 3.
$T_{TX-HF-DJ-DD}$	Tx Deterministic Jitter > 1.5 MHz	Not specified	0.15 (max)	UI	Deterministic jitter only. Refer to Note 3.
$T_{TX-LF-RMS}$	Tx RMS Jitter < 1.5 MHz	Not specified	3.0	ps RMS	Total energy measured over a 10-kHz to 1.5-MHz range.
$T_{TX-RISE-FALL}$	Tx Rise and Fall Time	0.125 (min)	0.15 (min)	UI	Measured differentially from 20 to 80% of swing. Refer to Note 3.
$T_{RF-MISMATCH}$	Tx Rise/Fall Mismatch	Not specified	0.1 (max)	UI	Measured from 20 to 80% differentially. Refer to Note 3.

Table 18-8. 2.5 and 5.0 GT/s PCI Express Transmitter (Signal Group a) – AC and DC Characteristics *(Cont.)*

Symbol	Parameter	2.5 GT/s	5.0 GT/s	Units	Comments
BW_{TX-PLL}	Maximum Tx PLL Bandwidth	22 (max)	16 (max)	MHz	Second Order PLL Jitter Transfer Bounding function. Refer to Note 5.
$BW_{TX-PLL-LO-3DB}$	Minimum Tx PLL Bandwidth for 3-dB Peaking	1.5 (min)	8 (min)	MHz	Second Order PLL Jitter Transfer Bounding function. Refer to Notes 5 and 7.
$BW_{TX-PLL-LO-1DB}$	Minimum Tx PLL Bandwidth for 1-dB Peaking	Not specified	5 (min)	MHz	
$PKG_{TX-PLL1}$	TX PLL Peaking with 8-MHz Minimum Bandwidth	Not specified	3.0 (max)	dB	
$PKG_{TX-PLL2}$	TX PLL peaking with 5-MHz Minimum Bandwidth	Not specified	1.0 (max)	dB	Refer to Note 7.
$RL_{TX-DIFF}$	TX Differential Return Loss (Package + Silicon)	10 (min)	10 (min) for 0.05 to 1.25 GHz 8 (min) for 1.25 to 2.5 GHz	dB	
RL_{TX-CM}	TX Common Mode Return Loss (Package + Silicon)	6 (min)	6 (min)	dB	S_{11} parameter. **2.5 GT/s** – Measured over 0.05- to 1.25-GHz range. **5.0 GT/s** – Measured over 0.05- to 2.5-GHz range.
$Z_{TX-DIFF-DC}$	DC Differential Tx Impedance	80 (min) 120 (max)	120 (max)	Ω	Tx DC Differential mode low impedance. Parameter is captured for 5.0 GHz by $RL_{TX-DIFF}$.
$V_{TX-CM-AC-PP}$	Tx AC Common Mode Voltage (5.0 GT/s)	Not specified	100 (max)	mVPP	Refer to Note 6.
$V_{TX-CM-AC-P}$	Tx AC Common Mode Voltage (2.5 GT/s)	20 (max)	Not specified	mVPP	Refer to Note 6.
$I_{TX-SHORT}$	Tx Short Circuit Current Limit	90 (max)	90 (max)	mA	Total current the Transmitter can provide when shorted to its ground.
$V_{TX-DC-CM}$	Tx DC Common Mode Voltage	0 (min) 3.6 (max)	0 (min) 3.6 (max)	V	Allowed DC common mode voltage, under any conditions.

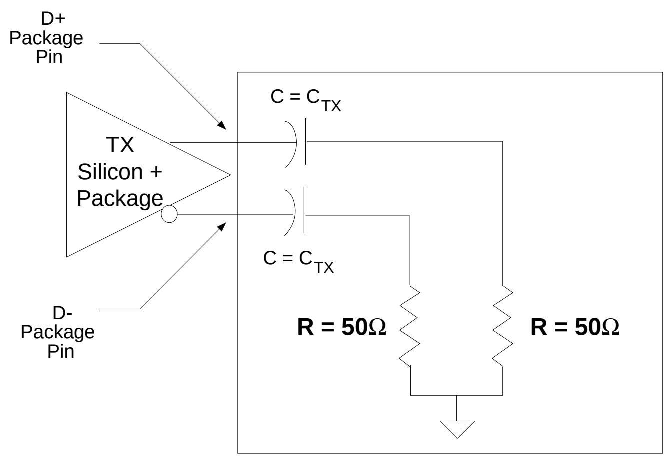
Table 18-8. 2.5 and 5.0 GT/s PCI Express Transmitter (Signal Group a) – AC and DC Characteristics *(Cont.)*

Symbol	Parameter	2.5 GT/s	5.0 GT/s	Units	Comments
$V_{TX\text{-}CM\text{-}DC\text{-}ACTIVE\text{-}IDLE\text{-}DELTA}$	Absolute Delta of DC Common Mode Voltage during L0 Link PM state and Electrical Idle	0 (min) 100 (max)	0 (min) 100 (max)	mV	$\|V_{TX\text{-}CM\text{-}DC}$ [during L0] $- V_{TX\text{-}CM\text{-}Idle\text{-}DC}$ [during Electrical Idle]$\| \leq 100$ mV $V_{TX\text{-}CM\text{-}DC} = DC_{(avg)}$ of $\|V_{TX\text{-}D+} + V_{TX\text{-}D-}\| / 2$ [L0] $V_{TX\text{-}CM\text{-}Idle\text{-}DC} = DC_{(avg)}$ of $\|V_{TX\text{-}D+} + V_{TX\text{-}D-}\| / 2$ [Electrical Idle]
$V_{TX\text{-}CM\text{-}DC\text{-}LINE\text{-}DELTA}$	Absolute Delta of DC Common Mode Voltage between D+ and D-	0 (min) 25 (max)	0 (min) 25 (max)	mV	$\|V_{TX\text{-}CM\text{-}DC\text{-}D+} - V_{TX\text{-}CM\text{-}DC\text{-}D-}\| \leq 25$ mV $V_{TX\text{-}CM\text{-}DC\text{-}D+} = DC_{(avg)}$ of $\|V_{TX\text{-}D+}\|$ $V_{TX\text{-}CM\text{-}DC\text{-}D-} = DC_{(avg)}$ of $\|V_{TX\text{-}D-}\|$
$V_{TX\text{-}IDLE\text{-}DIFF\text{-}AC\text{-}p}$	Electrical Idle Differential Peak Output Voltage	0 (min) 20 (max)	0 (min) 20 (max)	mV	$V_{TX\text{-}IDLE\text{-}DIFFp} = \|V_{TX\text{-}Idle\text{-}D+} - V_{TX\text{-}Idle\text{-}D-}\| \leq 20$ mV Voltage must be high-pass filtered, to remove any DC component.
$V_{TX\text{-}IDLE\text{-}DIFF\text{-}DC}$	DC Electrical Idle Differential Peak Output Voltage	Not specified	0 (min) 5 (max)	mV	$V_{TX\text{-}IDLE\text{-}DIFF\text{-}DC} = \|V_{TX\text{-}Idle\text{-}D+} - V_{TX\text{-}Idle\text{-}D-}\| \leq 5$ mV Voltage must be high-pass filtered, to remove any AC component.
$V_{TX\text{-}RCV\text{-}DETECT}$	Amount of Voltage Change Allowed during Receiver Detection	600 (max)	600 (max)	mV	Total amount of voltage change that a Transmitter can apply, to sense whether a Low-Impedance Receiver is present.
$T_{TX\text{-}IDLE\text{-}MIN}$	Minimum Time Spent in Electrical Idle	20 (min)	20 (min)	ns	Minimum time a Transmitter must be in Electrical Idle. Used by the Receiver to start looking for an Electrical Idle Exit after successfully receiving an Electrical Idle Ordered-Set (EIOS).
$T_{TX\text{-}IDLE\text{-}SET\text{-}TO\text{-}IDLE}$	Maximum Time to Transition to a Valid Electrical Idle after Sending an Electrical Idle Ordered-Set	8 (max)	8 (max)	ns	After sending the required EIOS, the Transmitter must meet all Electrical Idle specifications within this time. This is measured from the end of the last UI of the last EIOS to the Tx in Electrical Idle.
$T_{TX\text{-}IDLE\text{-}TO\text{-}DIFF\text{-}DATA}$	Maximum Time to Transition to Valid Differential Signaling after Leaving Electrical Idle	8 (max)	8 (max)	ns	Maximum time to transition to valid differential signaling, after leaving Electrical Idle. This is considered a de-bounce time to the Tx.
$T_{CROSSLINK}$	Cross-Link Random Timeout	1.0 (max)	1.0 (max)	ms	Random timeout that helps resolve potential conflicts in the cross-link configuration.
$L_{TX\text{-}SKEW}$	Lane-to-Lane Output Skew	500 ps + 2 UI (max)	500 ps + 4 UI (max)	ps	Static skew between any two Lanes within a single Transmitter.
C_{TX}	AC-Coupling Capacitor	75 (min) 200 (max)	75 (min) 200 (max)	nF	All Transmitters shall be AC-coupled. The AC coupling is required either within the media, or within the transmitting component itself.

Notes:

1. *SSC permits a +0, -5,000 ppm modulation of the clock frequency, at a modulation rate not to exceed 33 kHz.*
2. *Specified at the measurement point into a timing and voltage compliance test load, as illustrated in Figure 18-1.*

Figure 18-1. Compliance Test/Measurement Load



3. *Measurements at 5.0 GT/s require an oscilloscope with a bandwidth of ≥12.5 GHz, or equivalent, while measurements made at 2.5 GT/s require a scope with at least 6.2 GHz bandwidth. Measurements at 5.0 GT/s must de-convolve effects of the compliance test board, to yield an effective measurement at the Tx balls. 2.5 GT/s can be measured within 200 mils of the Tx device's balls; however, de-convolution is recommended. At least 10^6 UI of data must be acquired.*
4. *Transmitter jitter is measured by driving the Tx under test with a low jitter "ideal" clock and connecting the device under test (DUT) to a reference load.*
5. *The Tx PLL bandwidth must lie between the minimum and maximum ranges listed in Table 18-8. PLL peaking must lie below the values listed in Table 18-8.*

   The PLL bandwidth extends from zero (0) up to the value(s) specified in Table 18-8.
6. *Measurement is made over at least 10^6 UI.*
7. *A single combination of PLL bandwidth and peaking is specified for 2.5 GT/s implementations. For 5.0 GT/s, two 20 combinations of PLL bandwidth and peaking are specified to permit designers to make a tradeoff between the two parameters. If the PLL's minimum bandwidth is ≥8 MHz, then up to 3.0 dB of peaking is permitted. If the PLL's minimum bandwidth is relaxed to ≥5.0 MHz, then a tighter peaking value of 1.0 dB must be met. In both cases, the maximum PLL bandwidth is 16 MHz.*

Table 18-9. 2.5 and 5.0 GT/s PCI Express Receiver (Signal Group b) – AC and DC Characteristics

Symbol	Parameter	2.5 GT/s	5.0 GT/s	Units	Comments
UI	Unit Interval	399.88 (min) 400.12 (max)	199.94 (min) 200.06 (max)	ps	UI does not account for variations caused by SSC.
$V_{RX-DIFF-PP-CC}$	Differential Rx Peak-to-Peak Voltage for Common REFCLK Rx Architecture	0.175 (min) 1.2 (max)	0.125 (min) 1.2 (max)	V	$V_{RX-DIFF-PP} = 2 \times \lvert V_{RX-D+} - V_{RX-D-} \rvert$
T_{RX-EYE}	Receiver Eye Time Opening	0.40 (min)	N/A	UI	Minimum eye time at Rx pins to yield a 10^{-12} Bit Error Rate. Receiver eye margins are defined into a 2 x 50Ω reference load.
$T_{RX-TJ-CC}$	Maximum Rx Inherent Timing Error	N/A	0.40 (max)	UI	Maximum Rx inherent total timing error for common REFCLK Rx architecture. Refer to Note 1.
$T_{RX-DJ-DD-CC}$	Maximum Rx Inherent Deterministic Timing Error	N/A	0.30 (max)	UI	Maximum Rx inherent deterministic timing error for common REFCLK Rx architecture. Refer to Note 1.
$T_{RX-EYE-MEDIAN-to-MAX-JITTER}$	Maximum Time Delta between the Median and Deviation from the Median	0.3 (max)	Not specified	UI	
$T_{RX-MIN-PULSE}$	Minimum Width Pulse at Rx	Not specified	0.6 (min)	UI	Measured to account for worst Tj at 10^{-12} Bit Error Rate.
$V_{RX-MAX-MIN-RATIO}$	Minimum/ Maximum Pulse Voltage on Consecutive UI	Not specified	5 (max)	Ratio	Rx eye must simultaneously meet V_{RX-EYE} limits.
$BW_{RX-PLL-HI}$	Maximum Rx PLL Bandwidth	22 (max)	16 (max)	MHz	Second Order PLL Jitter Transfer Bounding function. Refer to Note 2.
$BW_{RX-PLL-LO-3DB}$	Minimum Rx PLL Bandwidth for 3-dB Peaking	1.5 min	8 (min)	MHz	
$BW_{RX-PLL-LO-1DB}$	Minimum Rx PLL Bandwidth for 1-dB Peaking	Not specified	5 (min)	MHz	
$PKG_{RX-PLL1}$	Rx PLL Peaking with 8-MHz Minimum Bandwidth	Not specified	3.0	dB	
$PKG_{RX-PLL2}$	Rx PLL Peaking with 5-MHz Minimum Bandwidth	Not specified	1.0	dB	

Table 18-9. 2.5 and 5.0 GT/s PCI Express Receiver (Signal Group b) – AC and DC Characteristics *(Cont.)*

Symbol	Parameter	2.5 GT/s	5.0 GT/s	Units	Comments
$RL_{RX-DIFF}$	Rx Differential Return Loss (Package + Silicon)	10 (min)	10 (min) for 0.05 to 1.25 GHz 8 (min) for 1.25 to 2.5 GHz	dB	Refer to Note 3.
RL_{RX-CM}	Common Mode Return Loss	6 (min)	6 (min)	dB	Refer to Note 3.
Z_{RX-DC}	Rx DC Single-Ended Impedance	40 (min) 60 (max)	40 (min) 60 (max)	Ω	Required Rx D+ and D- DC impedance (50Ω ±20% tolerance). Refer to Note 4.
$Z_{RX-DIFF-DC}$	DC Differential Rx Impedance	80 (min) 120 (max)	Not specified	Ω	Rx DC Differential mode impedance. Parameter is captured for 5.0 GHz by $RL_{RX-DIFF}$. Refer to Note 4.
$V_{RX-CM-AC-P}$	Rx AC Common Mode Voltage	150 (max)	150 (max)	mVP	Measured at Rx pins, into a pair of 50Ω terminations into Ground. Refer to Note 5.
$Z_{RX-HIGH-IMP-DC-POS}$	DC Input Common Mode Input Impedance for Voltage >0 during Reset or Power-Down	50K (min)	50K (min)	Ω	Rx DC common mode impedance with the Rx terminations not powered, measured over the range 0 to 200 mV (with respect to Ground). Refer to Note 6.
$Z_{RX-HIGH-IMP-DC-NEG}$	DC Input Common Mode Input Impedance for Voltage <0 during Reset or Power-Down	1.0K (min)	1.0K (min)	Ω	Rx DC common mode impedance with the Rx terminations not powered, measured over the range -150 to 0 mV (with respect to Ground). Refer to Note 6.
$V_{RX-IDLE-DET-DIFFp-p}$	Electrical Idle Detect Threshold	65 (min) 175 (max)	65 (min) 175 (max)	mV	$V_{RX-IDLE-DET-DIFFp-p} = 2 \times \lvert V_{RX-D+} - V_{RX-D-} \rvert$ Measured at the Receiver's package pins.
$T_{RX-IDLE-DET-DIFF-ENTERTIME}$	Unexpected Electrical Idle Enter Idle Detect Threshold Integration Time	10 (max)	10 (max)	ms	An un-expected Electrical Idle ($V_{RX-DIFFp-p} < V_{RX-IDLE-DET-DIFFp-p}$) must be recognized no longer than $T_{RX-IDLE-DET-DIFF-ENTERTIME}$ to signal an unexpected idle condition.
$L_{RX-SKEW}$	Total Lane-to-Lane Skew	20 (max)	8 (max)	ns	Across all Lanes on a Port. Includes variation in the length of a SKIP Ordered-Set at the Rx, as well as any delay differences arising from the interconnect itself. Refer to Note 7.

Notes:

1. *The four inherent timing error parameters are defined for the convenience of Rx designers, and they are measured during Receiver tolerancing.*
2. *Two combinations of PLL bandwidth and peaking are specified at 5.0 GT/s, to permit designers to make trade-offs between the two parameters. If the PLL's minimum bandwidth is $\geq$8 MHz, then up to 3.0 dB of peaking is permitted. If the PLL's minimum bandwidth is relaxed to $\geq$5.0 MHz, then a tighter peaking value of 1.0 dB must be met.*

   A PLL bandwidth extends from zero up to the value(s) defined as the minimum or maximum in Table 18-9. For 2.5 GT/s, a single PLL bandwidth and peaking value of 1.5 to 22 MHz and 3.0 dB are defined.
3. *Measurements must be made for both common mode and differential return loss. In both cases, the DUT must be powered up and DC-isolated, and its D+/D- inputs must be in the low-Z state.*
4. *The Rx DC single-ended impedance must be present when the Receiver terminations are first enabled, to ensure that the Receiver Detect occurs properly. Compensation of this impedance can start immediately, and the Rx single-ended impedance (constrained by $RL_{RX\text{-}CM}$ to $50\Omega \pm 20\%$) must be within the specified range by the time Detect is entered.*
5. *Common mode peak voltage is defined by the expression:*

   `max{|(Vd+ - Vd-) - V-CMDC|}`
6. *$Z_{RX\text{-}HIGH\text{-}IMP\text{-}DC\text{-}NEG}$ and $Z_{RX\text{-}HIGH\text{-}IMP\text{-}DC\text{-}POS}$ are defined, respectively, for negative and positive voltages at the input of the Receiver. Transmitter designers must comprehend the large difference between >0 and <0 Rx impedances when designing Receiver detect circuits.*
7. *The $L_{RX\text{-}SKEW}$ parameter exists to handle repeaters that re-generate REFCLK and introduce differing numbers of skips on different Lanes.*

Table 18-10. PCI Express Differential Clock Input (Signal Group c) – AC and DC Characteristics

Symbol	Parameter	Min	Typ	Max	Unit	Notes
F_{REFCLK}	Reference Clock Frequency			100	MHz	1
V_{SW}	Differential Voltage Swing (Peak-to-Peak)	125		1,200	mV	
DC_{REFCLK}	Input Clock Duty Cycle	40		60	%	
R_{TERM}	Input Parallel Termination (Differential)		100		Ω	
V_I	Input Voltage Range	-0.3		VDDA	V	
V_{IH}	Input High-Level Voltage	0.3		VDDA	V	
V_{IL}	Input Low-Level Voltage	-0.3		VDDA - 0.30	V	
V_{ID}	Input Differential Voltage	0.2		2.2	V_{PP}	
V_{CM}	Common Mode Voltage	0.25		VDDA - 0.25	V	
Z_{IN}	Differential Input Impedance	80		120	Ω	

Notes:

1. *PEX_REFCLKn/p do not require AC coupling capacitors, when driven from a High-Speed Current Steering Logic (HCSL) source. Use with other Clock driver types (such as LVDS or LVPECL) has not been characterized.*

Table 18-11. PEX_REFCLKOUT – DC Electrical Characteristics

Parameter	Parameter	Min	Typ	Max	Unit	Notes
I_{OH}	Output High-Level Current	11	14	20	mA	
I_{OL}	Output Low-Level Current	-11	-14	-20	mA	
I_{OZH}	Output High Leakage Current	-10	–	10	µA	
I_{OZL}	Output Low Leakage Current	-10	–	10	µA	

Table 18-12. PEX_REFCLKOUT – AC Electrical Characteristics

Parameter	Parameter	Min	Typ	Max	Unit	Notes
V_{OH}	High-Level Output Voltage (Single-Ended)	0.44	0.7 V	1.1	V	
V_{OL}	Low-Level Output Voltage (Single-Ended)	-0.06	0.0 V	0.04	V	
V_{CROSS}	Output Cross-Point Voltage	0.25	0.35	0.55	V	
T_{OR}	Output Rising Edge Rate	0.58	2.5	4.0	V/ns	
T_{OF}	Output Falling Edge Rate	0.58	2.5	4.0	V/ns	
V_{MAX}	Absolute Maximum Output Voltage, Including Overshoot	–	–	1.15	V	
V_{MIN}	Absolute Minimum Output Voltage, Including Undershoot	-0.3	–	–	V	

18.7 USB Electrical Specifications

18.7.1 USB Full- and High-Speed AC/DC Specifications

Table 18-13. USB Full-Speed AC Specifications (Signal Group f)

Symbol	Parameter	Conditions	Waveform	Min	Max	Units
T_{FR}	Full-Speed Rise Time	10% to 90%, C_L = 50 pF	Figure 18-2 Figure 18-3	4	20	ns
T_{FF}	Full-Speed Fall Time	90% to 10%, C_L = 50 pF	Figure 18-2 Figure 18-3	4	20	ns
T_{FRFM}	Differential Rise and Fall Time Matching	(T_{FR} / T_{FF}), Refer to Note 10	Figure 18-2 Figure 18-3	90	110	%
Z_{DRV}	Driver Output Resistance	Steady-State Drive		10	15	Ω
$T_{FDRATHS}$	Full-Speed Data Rate			11.994	12.006	Mbps
T_{DJ1}	Source Differential Driver Jitter to Next Transition	Refer to Notes 7, 8, 10, 12	Figure 18-4	-2	2	ms
T_{DJ2}	Source Differential Driver for Paired Transitions	Refer to Notes 7, 8, 10, 12	Figure 18-4	-1	1	ms
T_{FDEOP}	Source Jitter for Differential Transition to SE0 Transition	Refer to Notes 8, 11	Figure 18-5	-2	5	ms
T_{JR1}	Receiver Data Jitter Tolerance to Next Transition	Refer to Note 8	Figure 18-6	-18.5	18.5	ns
T_{JR2}	Receiver Data Jitter Tolerance for Paired Transitions	Refer to Note 8	Figure 18-6	-9	9	ns
T_{FEOPT}	Source SE0 Interval of EOP		Figure 18-5	160	175	ns
T_{FEOPR}	Receiver SE0 Interval of EOP	Refer to Note 13	Figure 18-5	82		ns
T_{FST}	Width of SE0 Interval during Differential Transition			14		ns

Table 18-14. USB Full-Speed DC Specifications (Signal Group f)

Symbol	Parameter	Conditions	Min	Max	Units
V_{IH}	Input High Level (Driven)	Refer to Note 4	2.0		V
V_{IHZ}	Input High Level (Floating)	Refer to Note 4	2.7	3.6	V
V_{IL}	Input Low Level	Refer to Note 4		0.8	V
V_{DI}	Differential Input Sensitivity	| (D+) - (D-) |	0.2		V
V_{CM}	Differential Common Mode Range	Includes VDI range	0.8	2.5	V
V_{OL}	Output Low Level	Refer to Notes 4, 5	0.0	0.3	V
V_{OH}	Output High Level (Driven)	Refer to Notes 4, 6	2.8	3.6	V
V_{SE1}	Single-Ended One		0.8		V
V_{CRS}	Output Signal Crossover Voltage	Refer to Note 10	1.3	2.0	V
C_{IO}	I/O Capacitance	Pin to Ground		20	pF

Table 18-15. USB High-Speed AC Specifications (Signal Group f)

Symbol	Parameter	Conditions	Min	Max	Units
T_{HSR}	High-Speed Rise Time	10% to 90%	500		ps
T_{HSF}	High-Speed Fall Time	90% to 10%	500		ps
	Driver Waveform Requirements	Specified by eye pattern template. Refer to the *USB r2.0*, Figure 7-17 (Template 5)			
Z_{DRV}	Driver Output Resistance	Steady-State Drive	10	15	Ω
T_{HSDRV}	High-Speed Data Rate		479.760	480.240	Mbps
	Data Source Jitter	Source and receiver jitter specified by eye pattern templates in the *USB r2.0*, Section 7.1.2.2			
	Receiver Jitter Tolerance				

Table 18-16. USB High-Speed DC Specifications (Signal Group f)

Symbol	Parameter	Conditions	Min	Max	Units
V_{HSSQ}	High-Speed Squelch Detection Threshold (Differential Signal Amplitude		100	150	mV
V_{HSDSC}	High-Speed Disconnect Detection Threshold (Differential Signal Amplitude)		525	625	mV
V_{HSCM}	High-Speed Data Signalling Common Mode Voltage Range		-50	500	mV
V_{HSOI}	High-Speed Idle Level		-10	10	mV
V_{HSOH}	High-Speeed Data Signalling High		360	440	mV
V_{HSOL}	High-Speed Data Signalling Low		-10	10	mV
V_{CHIRPJ}	Chirp J Level (Differential Voltage)		700	1100	mV
V_{CHIRPK}	Chirp K Level (Differential Voltage)		-900	-500	mV
C_{IO}	I/O Capacitance	Pin to Ground		20	pF

18.7.1.1 USB High/Full Speed AC/DC Specification Notes

The following notes are referenced by the tables provided in Section 18.7.1.

Notes:

1. *Measured at A plug.*
2. *Measured at A receptacle.*
3. *Measured at B receptacle.*
4. *Measured at A or B connector.*
5. *Measured with* R_L *of 1.425KΩ to 3.6V.*
6. *Measured with* R_L *of 14.25KΩ to Ground.*
7. *Timing difference between the Differential Data signals.*
8. *Measured at the crossover point of the Differential Data signals.*
9. *The maximum load specification is the maximum effective capacitive load allowed that meets the Target hub VBUS drop of 330 mV.*
10. *Excluding the first transition from the Idle state.*
11. *The two transitions should be a (nominal) bit time apart.*
12. *For both transitions of differential signaling.*
13. *Must accept as a valid EOP.*
14. *Single-ended capacitance of D+ or D- is the capacitance of D+/D- to all other conductors and, if present, shield in the cable. That is, to measure the single-ended capacitance of D+, short D-, VBUS, Ground, and the shield line together and measure the capacitance of D+ to other conductors.*
15. *For high-power devices (non-hubs), when enabled for Remote Wakeup.*

18.7.2 USB Full-Speed Port AC Waveforms

Figure 18-2. Data Signal Rise and Fall Time

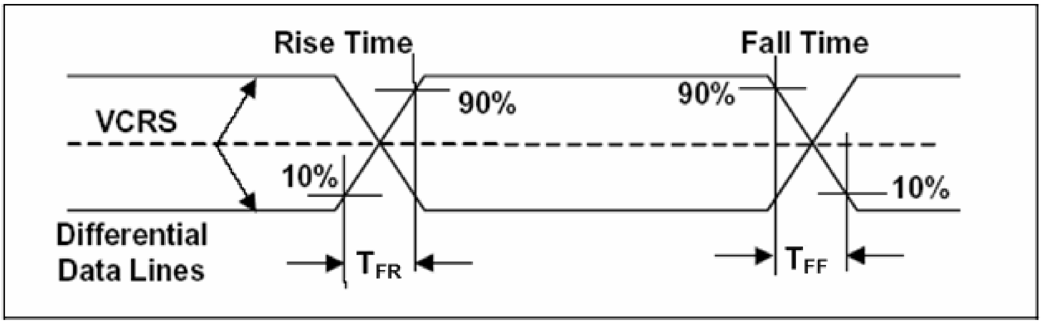


Figure 18-3. Full-Speed Load

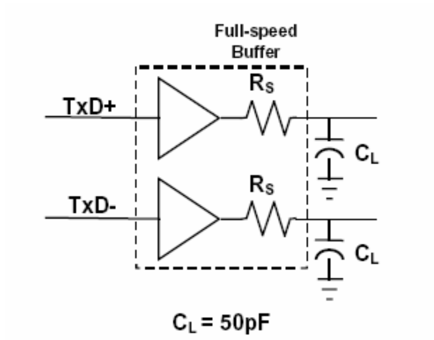


Figure 18-4. Source Differential Driver Jitter

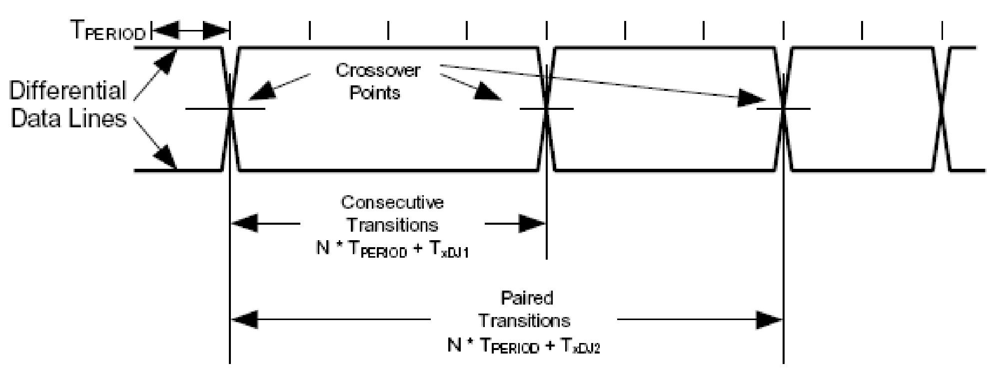


Figure 18-5. Differential to EOP Transition Skew and EOP Width

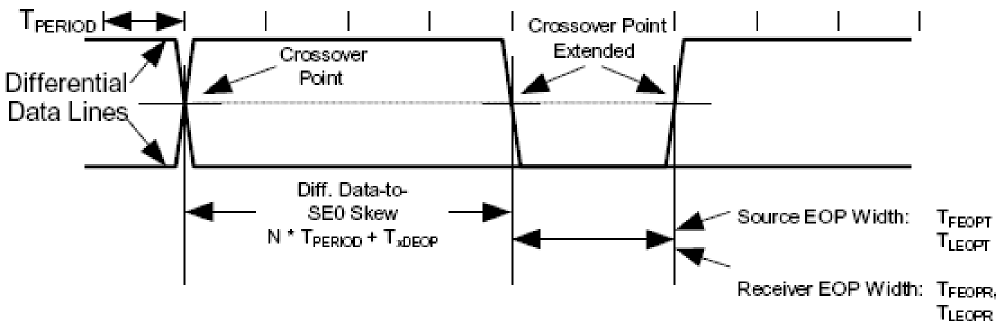
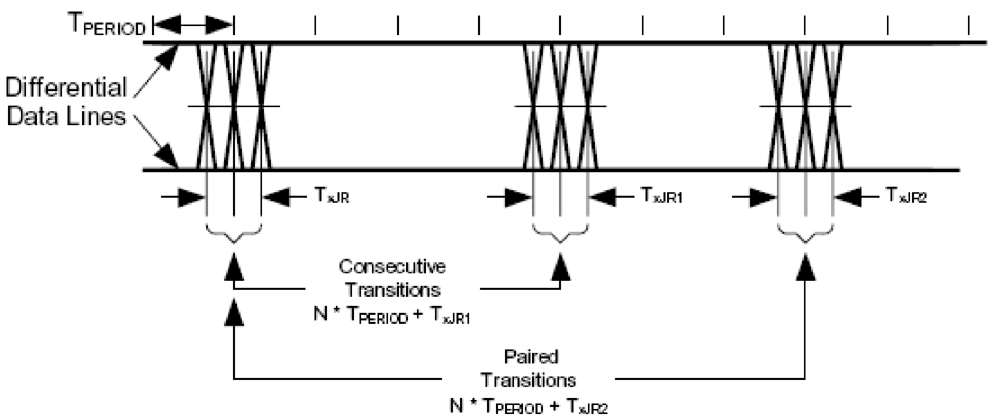


Figure 18-6. Receiver Jitter Tolerance



18.7.3 SuperSpeed USB Interface Specifications

The USB 3382 SuperSpeed USB interface is conformant to the *USB r3.0*. (Refer to the *USB r3.0* for further details.)

Table 18-17. SuperSpeed USB Transmitter Normative Electrical Parameters at 5.0 GT/s (Signal Group d)

Symbol	Parameter	Min	Max	Units	Comments
UI	Unit Interval	199.94	200.06	ps	The specified UI is equivalent to a tolerance of ±300 ppm for each device. Period does not account for SSC-induced variations.
$V_{TX-DIFF-PP}$	Differential p-p Tx Voltage Swing	0.4	1.2	V	Nominal is 1V p-p.
$V_{TX-DIFF-PP-LOW}$	Low-Power Differential p-p Tx Voltage Swing	0.4	1.2	V	Refer to the *USB r3.0*, Section 6.7.2. There is no de-emphasis requirement in this mode. De-emphasis is implementation-specific for this mode.
$V_{TX-DE-RATIO}$	Tx De-Emphasis	3.0	4.0	dB	Nominal is 3.5 dB.
$R_{TX-DIFF-DC}$	DC Differential Impedance	80	120	Ω	
$V_{TX-RCV-DETECT}$	Amount of Voltage Change Allowed during Receiver Detection	N/A	0.6	V	Detect voltage transition should be an increase in voltage on the pin looking at the Detect signal, to avoid a high-impedance requirement when an “off” Receiver’s input goes below ground.
$C_{AC-COUPLING}$	AC Coupling Capacitance	75	200	nF	All Transmitters should be AC-coupled. The AC coupling is required either within the media or within the transmitting component itself.
$T_{CDR\_SLEW\_MAX}$	Maximum Slew Rate		10	ms/s	

Table 18-18. SuperSpeed USB Transmitter Informative Electrical Parameters at 5.0 GT/s (Signal Group d)

Symbol	Parameter	Min	Max	Units	Comments
$t_{MIN-PULSE-DJ}$	Deterministic Minimum Pulse	0.96		UI	Tx pulse width variation that is deterministic.
$t_{MIN-PULSE-TJ}$	Tx Minimum Pulse	0.90		UI	Minimum Tx pulse at 10^{-12}, including Dj and Rj.
t_{TX-EYE}	Transmitter Eye	0.625		UI	Includes all jitter sources.
$t_{TX-DJ-DD}$	Tx Deterministic Jitter		0.205	UI	Deterministic jitter only assuming the Dual Dirac distribution.
$C_{TX-PARASITIC}$	Tx Input Capacitance for Return Loss		1.25	pF	Parasitic capacitance to Ground.
R_{TX-DC}	Transmitter DC Common Mode Impedance	18	30	Ω	DC impedance limits to guarantee receiver detect behavior. Measured with respect to AC Ground over a voltage of 0 to 500 mV.
$I_{TX-SHORT}$	Transmitter Short-Circuit Current Limit		60	mA	Total current that the Transmitter can source when shorted to Ground.
$V_{TX-DC-CM}$	Transmitter DC Common Mode Voltage	0	2.2	V	Instantaneous allowed DC common-mode voltages at the USB connector side of the AC-coupling capacitors.
$V_{TX-CM-AC-PP\_ACTIVE}$	Tx AC Common Mode Voltage Active		100	mV_{p-p}	Maximum mismatch from Txp + Txn for both time and amplitude.
$V_{TX-CM-DC-ACTIVE-IDLE-DELTA}$	Absolute DC Common Mode Voltage between U1 and U0		200	mV	
$V_{TX-IDLE-DIFF-AC-pp}$	Electrical Idle Differential Peak-to-Peak Output Voltage	0	10	mV	
$V_{TX-IDLE-DIFF-DC}$	DC Electrical Idle Differential Output Voltage	0	10	mV	Voltage must be low-pass filtered to remove any AC component. This limits the Common Mode error when resuming U1 to U0.
T_SDD22	Differential Output Return Loss		-8	dB	
T_SCC22	Common Mode Return Loss		-6	dB	

Table 18-19. SuperSpeed USB Receiver Normative Electrical Parameters at 5.0 GT/s (Signal Group e)

Symbol	Parameter	Min	Max	Units	Comments
UI	Unit Interval	199.94	200.06	ps	UI does not account for SSC-induced variations.
$R_{RX\text{-}DC}$	Receiver DC Common Mode Impedance	18	30	Ω	DC impedance limits are needed to guarantee Receiver detect. Measured with respect to ground over a voltage of 500-mV maximum.
$R_{RX\text{-}DIFF\text{-}DC}$	DC Differential Impedance	80	120	Ω	
$Z_{RX\text{-}HIGH\text{-}IMP\text{-}DC\text{-}POS}$	DC Input Common Mode Input Impedance for V>0 during Reset or Power-Down	25		KΩ	Rx DC common mode impedance with the Rx terminations not powered, measured over the range 0 to 500 mV, with respect to Ground. Refer to Note 1.
$V_{RX\text{-}LFPS\text{-}DET\text{-}DIFFp\text{-}p}$	LFPS Detect Threshold	100	300	mV	Below the minimum, is noise. Must wakeup above the maximum.

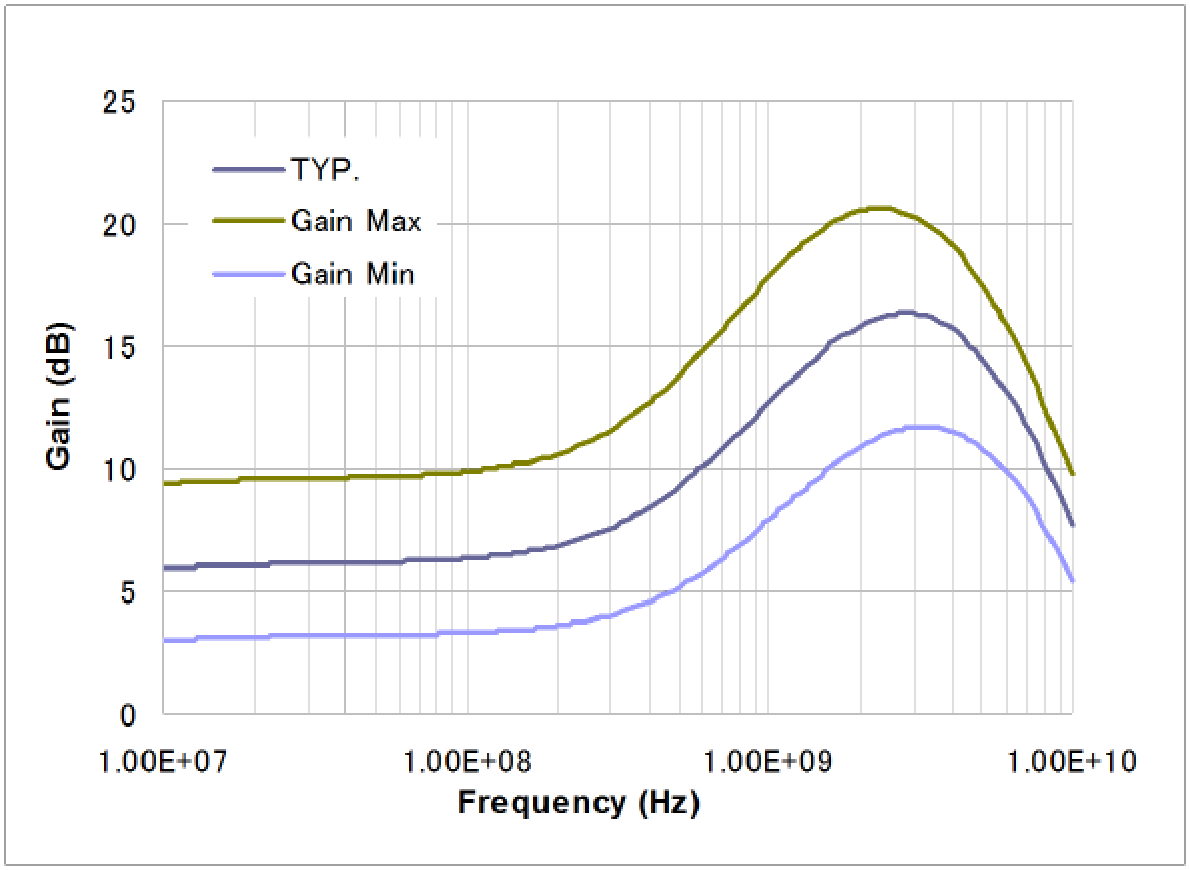
Notes:

1. *Only DC input CM input impedance for V > 0 is specified. DC input CM input impedance for V < 0 is not guaranteed, and could be as low as 0* Ω*.*

Table 18-20. SuperSpeed USB Receiver Informative Electrical Parameters at 5.0 GT/s (Signal Group e)

Symbol	Parameter	Min	Max	Units	Comments
$V_{RX\text{-}DIFF\text{-}PP\text{-}POST\text{-}EQ}$	Differential Rx Peak-to-Peak Voltage	30		mV	Measured after the Rx EQ function. Refer to the *USB r3.0*, Section 6.8.2.
$t_{RX\text{-}TJ}$	Max Rx Inherent Timing Error		0.45	UI	Measured after the Rx EQ function. Refer to the *USB r3.0*, Section 6.8.2.
$t_{RX\text{-}DJ\text{-}DD}$	Max Rx Inherent Deterministic Timing Error		0.285	UI	Maximum Rx inherent deterministic timing error.
R_TJ	Total Jitter		0.46	UI	
$C_{RX\text{-}PARASITIC}$	Rx Input Capacitance for Return Loss		1.1	pF	
$V_{RX\text{-}CM\text{-}AC\text{-}P}$	Rx AC Common Mode Voltage		150	mV peak	Measured at Rx pins into a pair of 50Ω terminations into Ground. Includes Tx and channel conversion, AC range up to 5 GHz.
$V_{RX\text{-}CM\text{-}DC\text{-}ACTIVE\text{-}IDLE\text{-}DELTA\_P}$	Rx AC Common Mode Voltage during the U1 to U0 Transition		200	mV peak	Measured at Rx pins into a pair of 50Ω terminations into Ground. Includes Tx and channel conversion, AC range up to 5 GHz.

Figure 18-7. 720700 EQ Transfer Function (EQ=2)



Chapter 19 Thermal and Mechanical Specifications

19.1 Thermal Characteristics

Table 19-1 lists sample thermal data for the USB 3382 at Industrial temperature (ambient temperature from -40 to +85°C).

Table 19-1. Sample Thermal Data[a]

Θ_{JA} (°C/W)			Psi_{jt}[b] (°C/W)	Θ_{JC} (°C/W)
0 m/s	1 m/s	2 m/s		
27.8	23.0	21.7	0.09	5.3

a. *Heat flow path (estimated):*
- *Heat dissipated from PCB – 80%*
- *Heat dissipated from package top – 8%*
- *Heat dissipated from others – 12%*

b. *Junction-to-top-bottom thermal characterization parameter. Used for estimating the junction temperature, by measuring TT in an actual environment.*

$Psi_{jt} = (TJ - TT) / PH$

where:

- *TT = Temperature at the bottom-center of the package*
- *PH = Power dissipation*
- *TJ = Junction temperature*

19.2 General Package Specifications

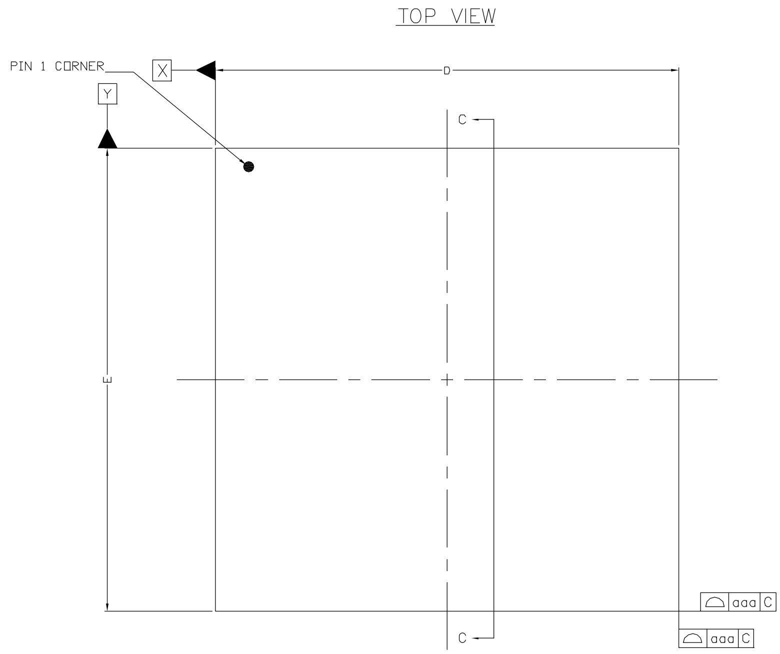
Table 19-2 lists general package specifications. For a more complete list, refer to Figure 19-1.

Table 19-2. General Package Specifications

Parameter	Specification
Package Type	Dual-Row QFN
Quantity of Pins	136 (staggered pins, including un-connected pins)
Package Dimensions	10 x 10 mm^2
Thermal Ground	4.940 x 4.940 mm^2 ±0.50 Center pad
Height	0.85 mm
Pitch	0.50 mm

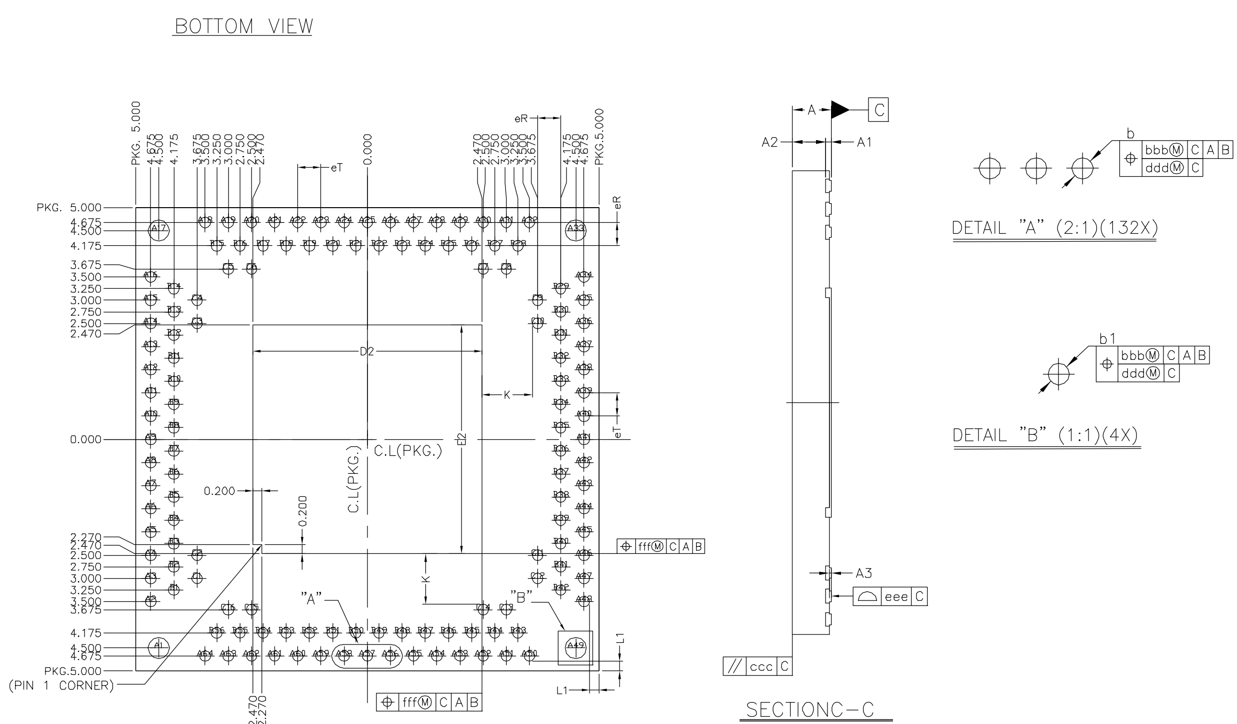
19.3 Mechanical Dimensions

Figure 19-1. Mechanical Dimensions (10 x 10 mm² Dual-Row QFN Package)



* CONTROLLING DIMENSION : MM

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	---	---	0.85	---	---	0.033
A3	0.020	0.050	0.080	0.0008	0.002	0.003
A2	0.640	0.675	0.710	0.025	0.027	0.028
A1	0.120	0.130	0.140	0.005	0.005	0.006
b	0.180	0.230	0.280	0.007	0.009	0.011
b1	0.410	0.460	0.510	0.016	0.018	0.020
D	10.00 BSC.			0.394 BSC.		
D2	4.890	4.940	4.990	0.193	0.194	0.196
E	10.00 BSC.			0.394 BSC.		
E2	4.890	4.940	4.990	0.193	0.194	0.196
eT	0.500			0.020		
eR	0.500			0.020		
K	1.040	1.090	1.140	0.041	0.043	0.045
L1	0.160	0.210	0.260	0.006	0.008	0.010
TOLERANCES OF FORM AND POSITION						
aaa	0.150			0.006		
bbb	0.100			0.004		
ddd	0.050			0.002		
ccc	0.100			0.004		
eee	0.080			0.003		
fff	0.100			0.004		



Appendix A General Information

A.1 Product Ordering Information

Contact your local PLX Sales Representative for ordering information.

Table A-1. Product Ordering Information

Part Numbers	Description
USB3382-AA50NI G USB3382-AB50NI G *where*	USB 3382 PCI Express Gen 2 to USB 3.0 SuperSpeed Peripheral Controller, 10 x 10 mm^2 136-pin Dual-Row QFN package USB – USB Express Product Family 3382 – Part Number AA, AB – Silicon Revision 50 – Signaling Rate (5.0 GT/s) N – QFN Package I – Industrial Temperature (if available) G – Lead-Free, ROHS 6/6- and Green-compliant Packaging (if available)
USB3382-AA-1D RDK USB3382-AB-1D RDK	USB 3382 Rapid Development Kit, configured as Root Complex with two x1 downstream facing PCI Express Ports
USB3382-AA-1U1D RDK USB3382-AB-1U1D RDK	USB 3382 Rapid Development Kit, configured as Add-in Card with one x1 upstream and one x1 downstream PCI Express Ports
USB3382-AA-2D RDK USB3382-AB-2D RDK	USB 3382 Rapid Development Kit, configured as Root Complex with one x2 downstream facing PCI Express Port
USB3382-AA-2U RDK USB3382-AB-2U RDK	USB 3382 Rapid Development Kit, configured as Add-in Card with one x2 upstream-facing PCI Express Port

A.2 United States and International Representatives and Distributors

PLX Technology, Inc., representatives and distributors are listed at www.plxtech.com.

A.3 Technical Support

PLX Technology, Inc., technical support information is listed at www.plxtech.com/support, or call 800 759-3735 (domestic only) or 408 774-9060.

单击下面可查看定价，库存，交付和生命周期等信息

>>Broadcom(博通)