KeyStone II Architecture
Universal Serial Bus 3.0 (USB 3.0)

User's Guide

Literature Number: SPRUHJ7A
April 2014–Revised August 2017

Contents

List of Figures

List of Tables

Preface

About This Manual

This User Guide describes the features, architecture, and details of the Universal Serial Bus 3.0 (USB 3.0) peripheral.

NOTE: The information in this document should be used in conjunction with information in the device-specific Keystone Architecture data manual that applies to the part number of your device.

Notational Conventions

This document uses the following conventions:

- Commands and keywords are in **boldface** font.
- Arguments for which you supply values are in *italic* font.
- Terminal sessions and information the system displays are in screen font.
- Information you must enter is in **boldface screen font**.
- Elements in square brackets ([ ]) are optional.

Notes use the following conventions:

NOTE: Means reader take note. Notes contain helpful suggestions or references to material not covered in the publication.

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

CAUTION

Indicates the possibility of service interruption if precautions are not taken.

WARNING

Indicates the possibility of damage to equipment if precautions are not taken.

Related Documentation from Texas Instruments

66AK2H14/12/06 Multicore DSP+ARM KeyStone II System-on-Chip (SoC) Data Manual	SPRS866
ARM CorePac User Guide for KeyStone II Device	SPRUHJ4
C66x CorePac User Guide	SPRUGW0
Multicore Programming Guide	SPRAB27
Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide	SPRUGV2
Power Management for KeyStone Devices	SPRABH0
Power Sleep Controller (PSC) for KeyStone Devices User Guide	SPRUGV4

Reference Documentation

Universal Serial Bus 3.0 Specification, Revision 1.0, USB-IF, November 12, 2008
On-The-Go Supplement to the USB 2.0 Specification, Revision 1.3, USB-IF, December 5, 2006
On-The-Go Supplement to the USB 2.0 Specification, Revision 2.0, May 8, 2009
Battery Charging Specification, Revision 1.2, November 2, 2010
PHY Interface for the PCI Express and USB 3.0 Architecture, Version 3.00, Intel Corp.
USB 2.0 Transceiver Macrocell Interface (UTMI) Specification, Revision 1.05, Intel Corp. March 29, 2001
UTMI+ Specification, Revision 1.0, ULPI Working Group, February 25, 2004
eXtensible Host Controller Interface For Universal Serial Bus (xHCI), Revision 1.0 with errata to 6/13/11, Intel Corp., June 13, 2011 (Request from USB-IF)

Introduction

NOTE: The information in this document should be used in conjunction with information in the device-specific Keystone Architecture data manual that applies to the part number of your device.

Topic **Page**

1.1 Purpose of the Peripheral

Similar to earlier versions of USB bus, USB 3.0 is a general-purpose cable bus, supporting data exchange between a host computer and a wide range of simultaneously accessible peripherals, but increases the data bandwidth up to 2*5Gbits/s bidirectionally by introducing a new operation mode – the Super-Speed (SS) mode – and a better power management scheme.

It is backward compatible with USB 2.0 by utilizing a dual-bus architecture, and is easy to use by preserving the model of smart host and simple device and leveraging the existing USB infrastructure.

1.2 Terminology Used in This Document

The following acronyms and abbreviations are used in the document.

Term	Definition
A-dev	A USB OTG device with an A-plug connected to its port (ID pin grounded). Default host
ACA	Accessory Charger Adaptor
ADP	Attach Detection Protocol: detects USB OTG attach/detach events.
API	Application Programming Interface (i.e. SW interface).
B-dev	A USB OTG device with a B-plug connected to its port (ID pin floating). Default peripheral.
BER	Bit Error Rate.
CDC	Communication Device Class: USB device class.
DDR	Dual Data Rate.
DMA	Direct Memory Access
DRD	Dual Role Device: USB Host and Peripheral capable.
DS	Down Stream (A USB port facing from a Host or Hub to a Device/Peripheral).
ECC	Error Correction Circuit
EOI	End of Interrupt
EOP	End of Packet
EP	Endpoint: USB communication channel between the USB link partners carrying a single transfer type (BULK, ISOCH, INT, or CONTROL) and bus sharing arbitration scheme.
FIFO	First-In First-Out.
FS	Full-Speed USB data rate (12 Mbps).
FSM	Finite State Machine.
HCD	Host Controller Driver, designates the USB SW.
HNP	Host Negotiation Protocol, OTG extension to swap USB host and peripheral roles.
HS	High-Speed USB data rate (480 Mbps).
IP	Intellectual Property.
IRQ	Interrupt Request.
ISOCH	Isochronous transfer type.
ISR	Interrupt Service Routing: SW procedure handling an IRQ.
ITP	Isochronous Timestamp Packet: USB SS micro-frame boundary packets.
J	Logical USB2 line level: Diff1 in HS/FS, Diff0 in LS. Idle FS/LS bus state.
K	Logical USB2 line level: Diff0 in HS/FS, Diff1 in LS. Resume bus state.
LFPS	Low-Frequency Periodic Signal.
LMP	Link Management Packet.
LS	Low-Speed USB data rate (1.5 Mbps).
LSB	Least Significant Bit.
MHz	Megahertz.
MMR	Memory Mapped Register.
MSB	Most Significant Bit.
MSC	Mass Storage Class.
OTG	On-The-Go extension to USB protocol.

Term	Definition
PHY	Physical Layer Device.
PIPE	PHY Interface for PCI Express, also used in USB3 SS PHY interface.
PIPE3	PIPE Interface for USB3 SS PHY interface.
POR	Power On Reset (cold reset).
RAM	Random Access Memory.
RX	Receive.
SDR	Single Data Rate.
SE	Single-Ended: USB state based on individual lines' state (D+/D-).
SE0	Single-Ended zero line state where D+=D-=0. Used for reset, FS/LS EOP.
SER	Soft Error Rate.
SOC	System On a Chip.
SOF	Start Of Frame.
SRP	Session Request Protocol. OTG extension to wake-up a system, allowing a B-device to request an A-device to turn on the VBUS power and start session.
SS	Super Speed
SSC	Spread Spectrum Clocking.
SW	SoftWare
TRB	Transfer Request Block.
TX	Transmit.
UAS	USB-Attached SCSI: ANSI standard for USB-attached storage.
ULPI	UTMI+ Low Pin count Interface.
US	Upstream facing from a device (or hub) to the host (or a hub).
USB	Universal Serial Bus.
USB IF	USB Implementers Forum. The governing organization that develops and maintains the USB specifications and compliance standard. www.usb.org
UTMI	USB 2.0 PHY Transceiver Macrocell Interface specification.
UTMI+	UTMI plus extensions to the UTMI spec. Among other improvements, it defines the PHY interface for OTG 2.0.
xHC	Host Controller, designates the USB HW that implements the xHCI specification.
xHCI	eXtensible Host Controller Interface. The specification that defines the register level interface for host controller.

1.3 Features

The USB 3.0 subsystem, supports the following USB features:

- Operation modes:
  - Supports USB 3.0 Host mode at SuperSpeed (SS, 5Gbps), High-Speed (HS, 480 Mbps), Full-Speed (FS, 12 Mbps), and Low-Speed (LS, 1.5 Mbps)
  - Supports USB 3.0 Device mode at SuperSpeed (SS, 5 Gbps), High-Speed (HS, 480 Mbps), and Full-Speed (FS, 12 Mbps)
  - Supports all modes of transfers - Control, Bulk, Interrupt, and Isochronous.
  - Support OTG 2.0 with support of HNP, SRP, and ADP.
- A DRD (Dual-Role-Device - Host or Device) USB controller with the following features:
  - Compatible to the xHCI specification in Host mode
  - Supports 15 IN (Receive), 15 OUT (Transmit) endpoints (EPs), and one EP0 endpoint which is bidirectional
  - Power management USB3.0 states for U0, U1, U2, and U3
  - Internal DMA controller
  - Descriptor caching and data pre-fetching used to meet system performance
  - Dynamic FIFO memory allocation for all endpoints

- Operation flexibility:
  - Same programming model for SS, HS, FS, and LS operation
  - Flexible stream allocation
  - Stream priority
  - Multiple interrupt lines:
    - 16 interrupts associated with 16 programmable Event Rings for multi-core support
    - An OABS interrupt line which combines all OTG, ADP, and SER events
    - A MISC interrupt line for all miscellaneous events
- External requirements:
  - Needs an external Charge Pump for VBUS 5 V generation
  - Needs an external reference clock input for USB PHY operation
  - Needs an external resistor for internal PHY termination calibration

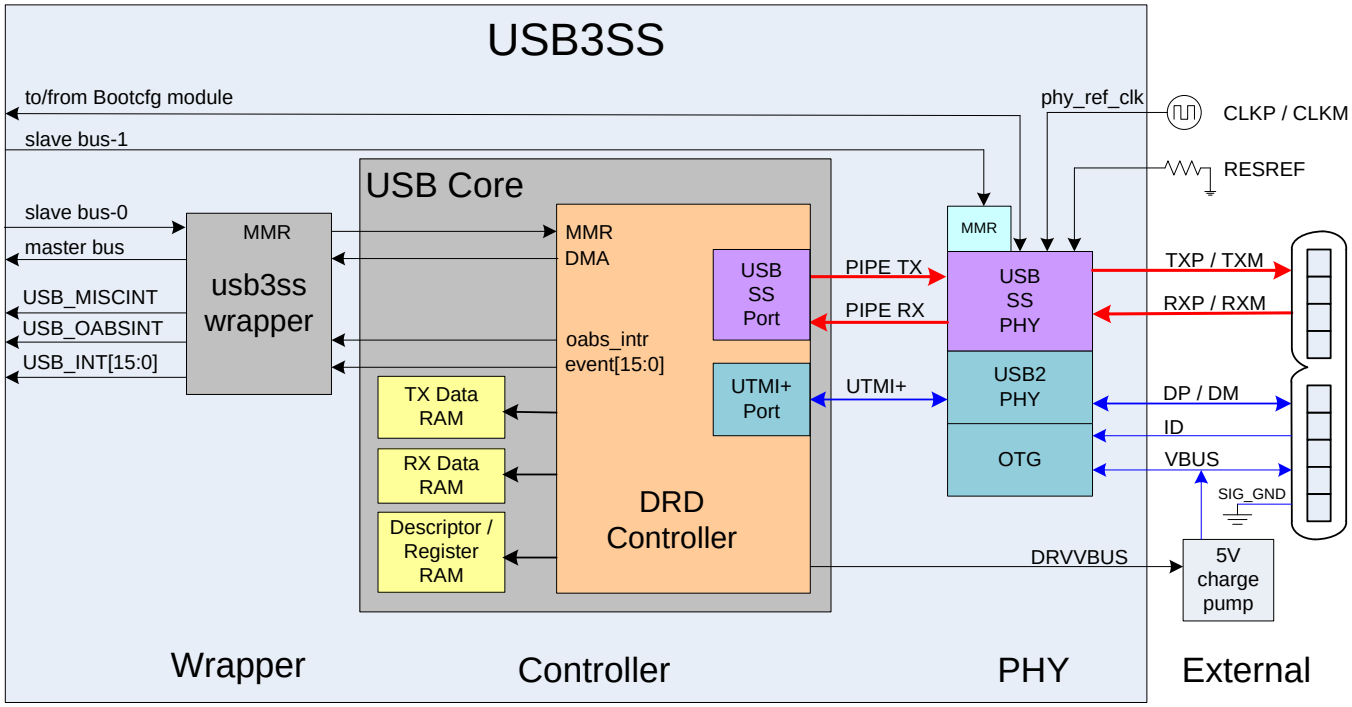
The following are USB features which are not supported:

- Battery Charger Support
- Accessory Charger Adaptor Support
- OTG 1.3 functionality
- OTG 3.0 functionality
- No Virtualization support.
- No Debug Class support.

1.4 Functional Block Diagram

Figure 1-1 shows the functional block diagram of the USB 3.0 subsystem (USB3SS), including a wrapper module, a USB controller module, a PHY module, external interfaces, and internal interfaces.

Figure 1-1. USB 3.0 Subsystem (USB3SS) Functional Block Diagram



- The wrapper module, which provides:
  - A bridge to the system bus
  - Local MMR to configure and control the USB controller
  - Interrupt management
  - Debug Interface to observe the state of internal signals.
  - Debug access to USB Core internal RAMs for the RX/TX data in FIFO, the cached parameters, as well as for observation of a selection of DRD internal signals
- The USB Core with a Dual-Role-Device Controller module and internal RAMs:
  - Compatible to the xHCI standard in Host mode
  - Support Host, Device, and OTG2.0 modes
  - Internal DMA controller for high-bandwidth low overhead data transfer
  - Local RAMs with ECC, supporting:
    - Dynamic Tx/Rx FIFO memory allocation for all EPs
    - Descriptor caching and data pre-fetching for high performance
  - Power management for USB3.0 states for U0, U1, U2, and U3
  - Supports all transfer types - Control, Bulk, Interrupt, and Isochronous
  - Supports high-bandwidth ISO mode
  - Supports 15 IN, 15 OUT endpoints (EPs), and one EP0 endpoint which is bidirectional
- The integrated USB PHY module:
  - A USB3 SS PHY connected to the core via a PIPE3 port
  - A USB2 PHY and a OTG2.0 PHY connected to the core via a UTMI+ (L3) port
- External interfaces (towards device boundary):
  - To USB Connector:
    - TXP/TXM - a differential signal pair for Super-Speed (SS) transmit
    - RXP/RXM - a differential signal pair for Super-Speed (SS) receive
    - DP/DM - a bidirectional signal pair for HS, FS, LS mode operation
    - VBUS - an analog input for monitoring the voltage on VBUS. It is also an analog output used for OTG ADP protocol: discharging and charging the capacitance on the VBUS line and detecting changes of rise time.
    - ID - an input signal with pullup for OTG A-device or B-device. The pullup is enabled and disabled automatically for sampling the ID pin
  - To an external Charge Pump for VBUS 5V generation:
    - DRVVBUS - an active high digital output that controls an external 5V charge pump. The core uses this digital output to automatically make sure that the VBUS has 5V applied by the charge pump during Host mode and in specific conditions during OTG mode.
  - To an external reference clock input for PHY operation:
    - CLKP/CLKM - a differential signal pair providing a reference clock
  - To an external reference resistor for training internal PHY terminations.
    - RESREF - a connection to ground through a 200M 1% resistor used for termination tuning and other internal analog references in the PHY.
- Internal interface (towards internal system):
  - A set of PHY control/status signals from/to the Bootcfg module, viewed as a group of system registers - USB_PHY_CTL
  - Slave Bus-1 - an interface for read and write to the PHY registers
  - Slave Bus-0 - an interface for read and write to the core’s control and status registers and debug access to the RAMs, as well as the Wrapper’s registers
  - Master Bus - an interface for USB core to access SoC system memory

  - Interrupt interface:
    - USB_INT[15:0] - 16 interrupts associated with 16 programmable Event Rings
    - USB_OABSINT - single interrupt line that combines OTG, ADP, and SER events
    - USB_MISCINT - single interrupt line for misc events

NOTE: The following information is device dependent. Users need to refer to the Data Manual of the specific device for details.

- The base address and initial value after reset of the USB_PHY_CTL registers
- The base address to the each of the two slave spaces - slave bus-0 and slave bus-1
- The map of interrupts - USB_MISCINT, USB_OABSINT, and USB_INT[15:0] to processor(s)
- The support for OTG mode

1.5 Supported Use Case Statement

This is a standard USB 3.0 module, and is optimized for following applications and systems:

- Portable electronic devices
- High-bandwidth applications

It supports all typical USB connections, and Table 1-1shows some examples.

Table 1-1. Typical Use Cases In Terms of Connections

Connectors (Receptacle)	Signals to Use	SS	HS/FS	LS (Host only)	OTG2.0	Comments
USB 3.0 micro-AB	TXP, TXM, RXP, TXM, DP, DM, VBUS, ID	Y	Y	Y	Y	Support Host, Device, and OTG.
USB 2.0 micro-AB	DP, DM, VBUS, ID	N	Y	Y	Y	Support Host, Device, and OTG (USB2.0 only).
USB 3.0 Type-A	TXP, TXM, RXP, TXM, DP, DM, VBUS	Y	Y	Y	N	Only used for Host (SS/HS/FS/LS).
USB 2.0 Type-A	DP, DM, VBUS	N	Y	Y	N	Support HS/FS/LS Host
USB 3.0 Type-B	TXP, TXM, RXP, TXM, DP, DM, VBUS	Y	Y	N	N	Support Device (SS, HS, FS, no LS), but not SS and HS/FS at the same time.
USB 2.0 Type-B	DP, DM, VBUS	N	Y	N	N	Only used for USB2.0 Device (no LS)

1.6 Industry Standard(s) Compliance Statement

This USB subsystem is fully-compliant with USB 3.0 standards and OTG v2.0 standards.

Architecture

This chapter describes the details of the USB3.0 peripheral's architecture.

Topic **Page**

2.1 Clock Control

The USB3SS contains several functional clock domains: BUS, PHYMMR, SS SUSPEND, USB2 SUSPEND, UTMI+ Interface, PIPE Interface, and RAM. An external clock is required to provide the reference clock for PHY operation.

Figure 1-1 shows these clock domains with different colors (the two suspend domains are small portions of the Controller, and are not shown out in the figure), as well as the PHY reference clock.

Table 2-1 provides the details of the clock drivers and the control of the clocks.

Table 2-1. USB3SS Clock Sources and Clock Control

Clocks	Min Freq	Max Freq.	Descriptions	Controls
bus_clk	60Mhz	400Mhz	The main bus clock to the wrapper module and to the DRD controller module. Driver is SYSCLK1 / 6.	Set SYSCONFIG.BUSCLKEN_N=0 to enable it.
phymmr_clk	10Mhz	200Mhz	The bus clock to the PHY MMR. The interface to PHY MMR is completely asynchronous. Driver is SYSCLK1 / 24.	Set SYSCONFIG.PHYMMRCLKEN_N=0 to enable it.
susp_clk	32Khz	125Mhz	The clock to a portion of DRD controller when the SS PHY is in its lowest power state (U3). The ADP logic is clocked with this clock too. Driver is SYSCLK1 / 24.	• Set GCTL.PwrDnScale = round up (susp_clk in khz / 16). • Set SYSCONFIG.SUSPCLKEN_N=0 to enable it.
ref_clk	16.5Mhz	125Mhz	The clock to a portion of DRD controller when USB2 PHY is in the suspended state. Driver is SYSCLK1 / 24.	• .Set SYSCONFIG.REFCLKEN_N=0 to enable it.
pipe_clk	125Mhz	125Mhz	PIPE3 interface clock. Derived from a off-chip reference clock - phy_ref_clk, and is shared with the USB2 PHY.	Controlled by USB_PHY_CTL4 and SYSCONFIG registers: • Set PHY_REF_USE_PAD to select amount two internal clocks source to the PHY boundary. To select the external clock USBREFCLK: – 1 on K2K device. – 0 on K2E and K2L devices. • Set PHY_REF_SSC_EN = 1b to enable Spread Spectrum (for SS mode, this must be enabled). • Set PHY_REF_SSP_EN = 1b to enable clock buffer to enable the clock (the ref clock to the PHY PLL) • Set FSEL[5:0] to match the frequency of the ref clock: – 100111b for 100Mhz (the default value) • Set SYSCONFIG.PIPECLKEN_N=0 to enable it.
utmi_clk	60Mhz	60Mhz	UTMI+ interface clock, sourced from a off-chip reference clock - phy_ref_clk and is shared with the SS PHY.	• To share the PHY reference clock with SS PHY, set: • USB_PHY_CTL4.PHY_REF_REFCLKSEL[1:0]=10b • Set SYSCONFIG.UTMICLKEN_N=0 to enable it.
ram_clk	30Mhz	400Mhz	RAM clock. It comes from either the bus_clk , or the bus_clk/2, or the pipe_clk.	Determined by the GCTL.RAMClkSel: • 0 - bus_clk • 1 - pipe_clk • 2 - pipe_clk/2 • 3 - reserved

Table 2-1. USB3SS Clock Sources and Clock Control (continued)

Clocks	Min Freq	Max Freq.	Descriptions	Controls
phy_ref_clk	N/A	N/A	An off-chip USB PHY reference clock used by a DPLL inside the PHY to generate the pipe_clk clock and the utmi_clk clock for PHY operation. Since the SSC modulation should not present in USB2 operation, and the SSC modulation for pipe_clk is generated internally, the phy_ref_clk clock must be a fixed-frequency clock	The supported shareable values of this reference clock on KeyStone II devices are: • 100 Mhz (TI recommends clock value) For programming selection of it, see the controls for pipe_clk and utmi_clk..

NOTE: Even thought multiple value can be selected for the external clock input to the phy_ref_clk, TI recommends customers to use only the 100 Mhz clock on their board design for KeyStone II devices.

2.2 Memory Map

Three blocks of USB3SS resources are memory mapped at the system level - the USB3SS control space, the PHY registers, and the 6 PHY control registers. The mappings are device dependent, and the Table 2-2 shows an example for the device - 66AK2H14/12/06.

Table 2-2. Memory Map of USB3SS Resources at System Level

USB3SS Resources	66AK2H12/06	
	Start	Bytes
USB3SS control space	0x02630000	128 KBytes
PHY Registers	0x02680000	512 KBytes
USB_PHY_CTL0 - USB_PHY_CTL5	0x02620738	24 Bytes

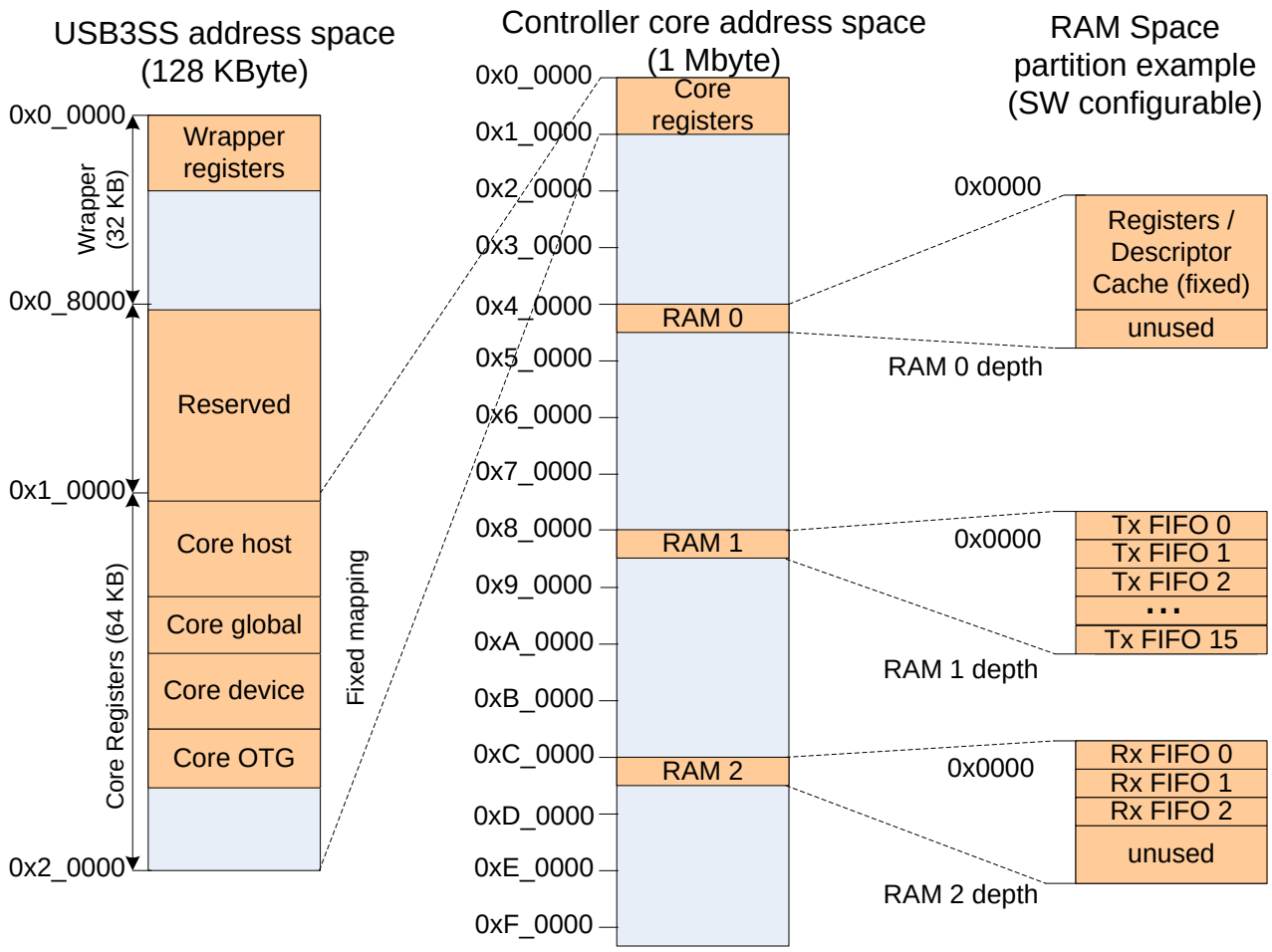
The 128 Kbytes USB3SS control space is further split into three blocks as shown in Table 2-3.

Table 2-3. Partitioning of USB3SS Internal Space

Block Name	Offset	Size	Comment
Wrapper Registers	0x0_0000	32 KBytes	Static mapping. Contains the registers for configuration, SER, interrupt interface, power management, etc.
Core Debug View	0x0_8000	32 KBytes	Dynamic page mapping via MMRAM_OFFSET register. Access to the controller core's entire 1M byte memory space.
Controller Core Space	0x1_0000	64 KBytes	Static mapping. Access to the cores' status and control registers only (no RAMs).

Figure 2-1 shows the address partition and re-mapping of the USB3SS control space.

Figure 2-1. Internal Memory Mapping of USB3SS



All Controller's functional registers are inside a single 64 KB block, and statically mapped inside the USB3SS control space (0x1_0000 to 0x1_FFFC). Normal operation does not require access to the rest of controller's core memory space.

The Controller's core space also covers the three internal RAMs, which holds the cache of data and descriptors (read out of or to be written to the main system memory by the Controller's built-in DMA) as well as some USB core registers (accessed by SW through the core register block). The RAM contents are never accessed directly for normal operation, only for debug purposes.

The data cache is composed of a number of Tx FIFOs and Rx FIFOs, mapped in RAM1 and RAM2, respectively. The mapping of each FIFO is software programmable through the GTXFIFOSZ/GRXFIFOSIZ registers, by a base address and FIFO depth configuration pair. Both parameters are expressed in FIFO's physical addresses, i.e., 64-bit word addresses, with the RAM's base at address 0. It is the programmer's responsibility to ensure that FIFOs are within the implemented RAM size and do not overlap each other.

The Controller's core register space is partitioned further into regions as in Table 2-4:

Table 2-4. Regions of Controller's Core Register Space

Register Region Name	Base Address	Size (KBytes)
xHCI Capability registers (Host)	0x00010000	0x0000020
xHCI Operational register (Host)	0x00010020	0x0000400
xHCI Port registers (Host)	0x00010420	0x0000020
xHCI Runtime registers (Host)	0x00010440	0x0000020

Table 2-4. Regions of Controller's Core Register Space (continued)

Register Region Name	Base Address	Size (KBytes)
xHCI Interrupter registers (Host)	0x00010460	0x00000020
xHCI Doorbell registers (Host)	0x00010480	0x00000400
xHCI Extended Capabilities registers	0x00010880	0x00000010
xHCI Supported Protocol Capability (USB 2.0)	0x00010890	0x00000010
xHCI Supported Protocol Capability (USB 3.0)	0x000108A0	0x00000010
Global registers	0x0001C100	0x00000600
Device registers	0x0001C700	0x00000500
USB 2.0 OTG and Battery Charger registers	0x0001CC00	0x00000040

For mappings of specific registers, see Chapter 3.

2.3 Signal Descriptions

The names of the pins used by the USB3SS with descriptions are shown in Table 2-5 through Table 2-7.

Table 2-5. USB Signal Pins Description

Pin Name	IOZP	IPD/IPU	Nominal Voltage	Description
USBDP	IOZ		3.3V	USB 2.0 and 1.1 specification-compliant signal pins. They are HS/FS/LS bidirectional differential data pins
USBDM	IOZ		3.3V	
USBTXP	I		0.85V	SuperSpeed TX- / TX+ pins. They are differential transmit lines, supporting 5 Gbps data rate. When connecting to a standard connector, the appropriate magnetics are recommended. AC coupling is required and always placed on the TX side and none on RX side. AC Coupling specified in the USB3.0 specification with a recommended range of 75nF-200nF for USB3 Gen 1 applications. Placement and value is best determined by simulation and modeling.
USBTXM	I		0.85V	
USBRXP	I		0.85V	SuperSeepd RX- / RX+ pins. They are differential receive lines, supporting 5 Gbp data rate. When connecting to a standard connector, the appropriate magnetics are recommended. AC coupling is required (always placed on the TX side)
USBRXM	I		0.85V	
USBCLKM	I		0.85Vp	The two USB PHY reference differential input clock pins.
USBCLKP	I		0.85Vp	

Table 2-6. USB Control, Configuration, and Monitor Signal Pins

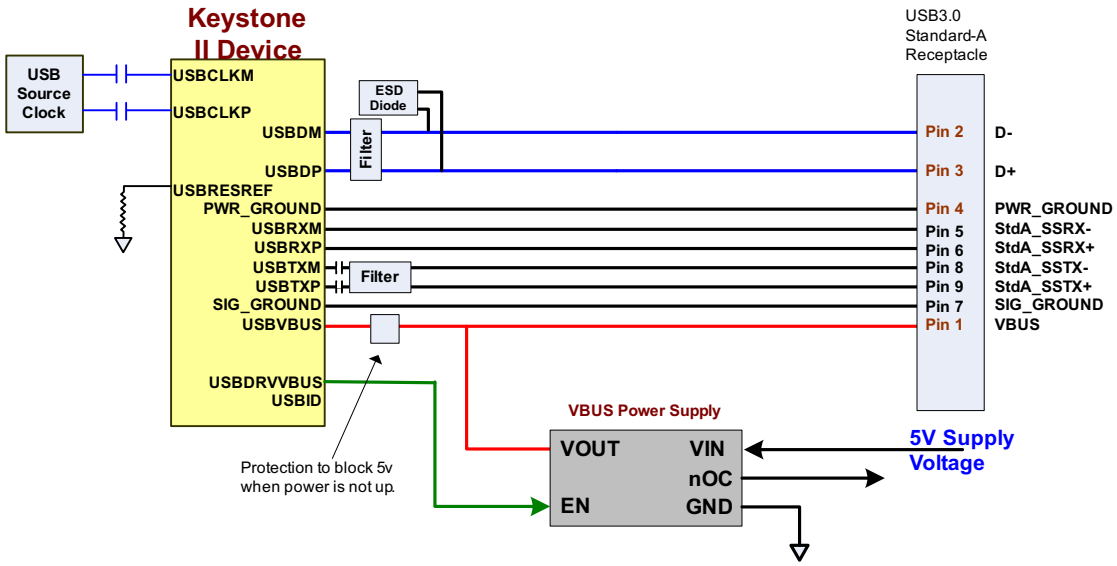
Pin Name	IOZP	Nominal Voltage	Description
USBDRVVBUS	O		A digital output signal for VBUS Power Supply Enabling. Used to enable an external charge pump to supply the +5V power to the VBUS port of the device as well as the VBUS port of the USB receptacle, when appropriate per the USB standard. Output is 0 at reset. The voltage value is device dependent, for K2K, K2E, and K2L, it is 1.8V
USBVBUS	A	5.0V	An analog input for monitoring the voltage on VBUS. It is also an analog output for OTG ADP protocol - discharging and charging the capacitance on the VBUS line and detecting changes of rise time.
USBID	A	3.3V	ID Pin with internal pull-up.
USBRESREF	P	0.25V	Termination Training Resistor. It is intended to connect an external resistor to ground for internal PHY termination calibration

Table 2-7. USB Power Signal Pins

Pin Name	IOZP	Nominal Voltage	Description
DVDD33	A	3.3V	Hi-Voltage Supply pin.
VDDUSB	A	0.85V	Low-Voltage Supply pin.
VPH	A	3.3V	SuperSpeed High Voltage Supply pin.
VP	A	0.85V	SuperSpeed Analog & Digital PHY Supply pin.
VPTX	A	0.85V	PHY Transmit Supply pin.

Figure 2-2 shows an example of typical USB3.0 pin connection as a Host with charge pump of KeyStone II device. For more details, refer to the KeyStone II USB Application Note.

Figure 2-2. Typical Pin Connection of USB 3.0 of KeyStone II Device



2.4 Pin Multiplexing

There is no pin muxing at device level for any pins used for USB.

2.5 Protocol Description(s)

The USB3SS supports USB 3.0 Host, USB 3.0 Device, and OTG 2.0 mode operations, and is fully compliant with USB Standard. For details of USB protocol, please refer to the USB standard Specifications.

From programming point of view, the eXtensible Host Controller Interface for USB (see *eXtensible Host Controller Interface for USB (xHCI) revision 1.0 with errata (errata files 1.7)*) is a standardized interface between a USB host controller (HC) and system software (HCD, Host Controller Driver), which supports SS, HS, FS, and LS of USB operation.

The USB3SS USB Controller is fully compliant with xHCI Specification when working in Host mode. For details of how USB3SS works in Host mode, refer to the xHCI Specification.

However, the USB Device mode operation is not standardized. The device mode controller is implemented by re-using some xHC logic. In both modes, the device TRB (Transfer Request Descriptors) as well as Event Buffer Rings are mapped in main memory to support the communication between application software and the Controller hardware.

Following sections will focus on how the DRD Controller supports the Device mode. More specifically, the data structure of Device TRBs and Event Buffer.

2.5.1 Device TRBs

Multiple Transfer Request Blocks (TRBs) are mapped in main memory consecutively to make up a Transfer Descriptor (TD) to define one data transfer transaction. Each TRB describes a data buffer which locates in main memory too. Initially created by the software, they are read by the controller DMA and updated by it, depending on USB activities.

First written to system memory by the software and then a pointer to the start of the TRB list is passed to the controller. The controller then owns the TD and software must not mess with it. Each TRB has one HWO (hardware owned) bit that is set by software before passing the TD to the controller. This bit specifies that the hardware (controller) owns the TRB. Once the controller has processed the TRB, it will change this bit to 0 indicating it has been processed and software can own it and modify it again to define new data transfers.

Even though device TRBs are inspired by and similar to the host TRBs defined by the *Extensible Host Controller Interface (xHCI) for USB* specification, they are not identical. The main difference is that device TRBs can be updated (written) by the controller DMA, unlike host TRBs, which are written only by the software driver and are read-only for the controller DMA.

NOTE: TRBs are data structures written to system memory by software to provides instructions to the controller. The controller similarly writes TRBs (called Event Buffer) in system memory to pass information back to software.

Table 2-8 through Table 2-12 describe the 128 bits of the TRB as four 32-bit words.

Table 2-8. Device TRB Word Structure

Name	PARAM_LO	PARAM_HI	STATUS	CONTROL
Address Offset	0x0	0x4	0x8	0xC

Figure 2-3. PARAM_LO Word of Device TRB Structure

31	0
BPTRL	
R/W - 0x00000000	

Legend: R = Read only; W = Write only; -n = value after reset

Table 2-9. PARAM_LO Word of Device TRB Structure

Bit	Field	Description
31-0	BPTRL	Buffer pointer low: Data buffer's base address pointer, lower

Figure 2-4. PARAM_HI Word of Device TRB Structure

31	0
BPTRH	
R/W - 0x00000000	

Legend: R = Read only; W = Write only; -n = value after reset

Table 2-10. PARAM_HI Word of Device TRB Structure

Bit	Field	Description
31-0	BPTRH	Buffer pointer high: Data buffer's base address pointer

Figure 2-5. STATUS Word of Device TRB Structure

31 28	27 26	25 24	23 0
TRBSTS	Reserved	PCM1	BUFSIZ
RW - 000	R - 00	RW - 00	RW - 0x000000

Legend: R = Read only; W = Write only; R/W = Read/Write; -n = value after reset; -x, value is indeterminate — see the device-specific data manual

Table 2-11. STATUS Word of Device TRB Structure

Bit	Field	Description
31-28	TRBSTS	TRB Status. Updated by hardware transfer status information before releasing the TRB. • 0x0 = OK • 0x1 = MissedIsoc: Isochronous interval missed or incomplete. • 0x2 = SetupPending: Another SETUP was received during current control transfer data/status phase.
27-26	Reserved	
25-24	PCM1	Packet count minus 1: Total number of packets in the buffer RW 0x0 descriptor for high-speed, high-bandwidth isochronous IN endpoints in an Isoc-First TRB. • 0x0 = 1 packet • 0x1 = 2 packets • 0x2 = 3 packets • 0x3 = 4 packets
23-0	BUFSIZ	Buffer size. Remaining size in bytes (to be sent/which can be RW 0x000000 received) in the TRB data buffer. Decremented by hardware after data is transferred. • 0x000000 = No remaining bytes in buffer • 0xFFFFFF = (16 MB –1) remaining byte in buffer (maximum size)

Figure 2-6. CONTROL Word of Device TRB Structure

31 30	29 14	13 12	11	10	9 4	3	2	1	0
Reserved	SID_SOFN	Reserved	IOC	IS_IMI	TRBCTL	CSP	CHN	LST	HWO
R - 0	RW - 0x0000	R - 0	R-0	R-0	R-0x0	R-0	R-0	R-0	R-0

Legend: R = Read only; W = Write only; R/W = Read/Write; -n = value after reset; -x, value is indeterminate — see the device-specific data manual

Table 2-12. CONTROL Word of Device TRB Structure

Bit	Field	Description
31-30	Reserved	
29-14	SID_SOFN	Stream ID/SOF number. • Stream-based bulk EP: The stream ID of the transfer. • Isochronous EP: (micro)frame number in which the last packet of the buffer is transmitted (debug).
13-12	Reserved	
11	IOC	Interrupt On Complete. Used to generate an interrupt in the middle (LST = 0) of a multi-TRB TD so that software can reclaim memory in the TR. Note: short packets have separate rules. Read-only for the USB controller DMA (hardware) • Read 0 = No action • Read 1 = XferInProgress event issued (with IOC bit set) on completion of buffer transfer.
10	IS_IMI	Interrupt on short packet/interrupt on Missed ISOCH. Controls the generation of the XferInProgress event. Read-only for the USB controller's DMA (hardware). • Read 0 = No action • Read 1 = XferInProgress event when: – Receiving an unexpected short packet (OUT EP, LST = 0, CSP = 1) or – The missed Isoch status is given (isochronous EP).

Table 2-12. CONTROL Word of Device TRB Structure (continued)

Bit	Field	Description
9-4	TRBCTL	TRB control. Type of TRB. Read-only for the USB controller DMA (hardware) • Read 0x01= Normal (Bulk/Interrupt/Control-Data-n with n1) • Read 0x02= Control-Setup • Read 0x03 = Control-Status-2 • Read 0x04 = Control-Status-3 • Read 0x05 = Control-Data-1 • Read 0x06 = Isochronous-1 (First TRB of Service Interval) • Read 0x07 = Isochronous-n (with n1) • Read 0x08 = Link TRB
3	CSP	Continue on short packet. Reaction of an OUT endpoint on reception of a short packet. Read-only for the USB controller DMA (hardware) • Read 0 = Generate XferComplete event and remove the stream. • Read 1 = Continue to the next buffer descriptor.
2	CHN	Chain buffers. Associates this TRB with the next in the same buffer descriptor. Always 0 in the last TRB of a buffer descriptor. Read-only for the USB controller DMA (hardware) • Read 0 = End of the current buffer descriptor. • Read 1 = Next TRB is part of the same buffer descriptor
1	LST	Last TRB. Indicates the last TRB in a list, that is, in a transfer for the endpoint/bulk-stream, which will cause an Xfercomplete interrupt. Read-only for the USB controller DMA (hardware) • Read 0 = TRB list not complete • Read 1 = Last TRB
0	HWO	Hardware owner. TRB ownership. Set to 1 by software when creating the TRB. Software cannot modify the TRB again until cleared to 0 by hardware. There are exceptions for short packets on OUT endpoints and link TRBs. • 0 = Software-owned TRB • 1 = Hardware-owned TRB

2.5.2 Device Mode Endpoint Commands

This subsection describes the endpoint commands passed by the device controller driver (software) to the device controller (hardware) through the appropriate memory-mapped control and status registers. The command is passed to physical EP #n through register USB_DEPCMD[n], with additional parameters passed through USB_DEPCMDPAR0[n], USB_DEPCMDPAR1[n] and USB_DEPCMDPAR2[n]. The number of parameters depends on the command, and the parameter registers must be written before the cmd register.

Table 2-13 provides a summary of the EP commands with which parameter registers are used (Y) in each case. Unused parameter registers are reserved.

Table 2-13. Device Mode Endpoint Commands Summary

	Command	Command Full Name	Parameter	PAR0	PAR1	PAR2
1	DEPCFG	Set EP configuration	64-bits	Y	Y	
2	DEPXFERCFG	Set EP transfer resource configuration	32-bits	Y		
3	DEPGETDSEQ	Get data sequence number	none			
4	DEPSETSTALL	Set stall	none			
5	DEPCSTALL	Clear stall	none			
6	DEPSTRTXFER	Start transfer	64-bits	Y	Y	
7	DEPUPDXFER	Update transfer	none			
8	DEPENDXFER	End transfer	none			
9	DEPSTARTCFG	Start new configuration	none			

NOTE: If GUSB2PHYCFG.bit6 is set to 1, it must be set to 0 prior to issuing any one of these commands and may be set back to 1 after the command completes.

2.5.2.1 Set EP Configuration (DEPCFG) Command

This command sets the physical endpoint configuration information.

The USB specification refers to EPs 0 through 15 with a direction of IN or OUT. The number here is called USB EP number, or Logical EP number. Contrastively, the index for the register set - DEPCMD[n] is called Physical EP number.The device controller supports a flexible mapping of Physical EP to Logical EP and this Command does the mapping.

This flexible EP mapping saves resources and allows you to use any combination of USB EPs. For example, when only USB EP4_IN and EP6_OUT are used in a USB application, the software doesn't need to allocate FIFO spaces for those un-used EPs.

Parameter setting:

- DEPCMD.CommandParam - unused (when USB_DEPCMD[n] is written)
- Additional parameters: 64-bits, stored in
  - USB_DEPCMDPAR0[n] - see details in the following tables
  - USB_DEPCMDPAR1[n] - see details in the following tables
  - USB_DEPCMDPAR2[n] - is reserved

Table 2-14. USB_DEPCMDPAR2 Fields for DEPCFG Command

Bit	Field	Description
31-0		Set to EP state when the Config field of Parameter 0 is 1. Otherwise, this is reserved.

Table 2-15. USB_DEPCMDPAR1[n] Fields for DEPCFG[n] Command , here n = Physical EP #

Bit	Field	Description
31	FIFOBASED	FIFO-based. Set to 1 if this isochronous EP represents a FIFO-based data stream where TRBs have fixed values and are never back by the core.
30	BULKBASED	Bulk-based. Set to 1 for an isochronous EP to ignore intervals programmed in the TRBs, like a bulk EP. 0: DEFAULT Normal behaviour for isochronous EP 1: BBISOC Bulk-based isochronous EP
29-25	EPNUMBER	USB Endpoint Number (Note: this is the USB logical EP #, instead of the Physical EP #) • bit[29:26] - Endpoint number • bit[25] - Endpoint direction – 0 - OUT – 1 - IN Physical EP0 must be allocated for control endpoint 0 OUT. Physical EP1must be allocated for control endpoint 0 IN.
24	STRMCAP	Stream Capable - indicates this EP is stream-capable (MaxStreams != 0) • 0: EP is not stream-capable • 1: EP is stream-capable
23-16	BINTERVAL_M1	bInterval_m1 Set to the bInterval value minus one. The valide values for this field are 1 through 13. The bInterval value is reported in the endpoint descriptor. When the core is operating in FS mode, this field must be set to 0.
15	LimitOutstandingTxDMA	Not applied to device that does not support the External Buffer Control (EBC) mode.
14	Reserved	
13	STREAMEVTEN	Stream Event DEPEVT enable • 0: event disabled • 1: event enabled

Table 2-15. USB_DEPCMDPAR1[n] Fields for DEPCFG[n] Command , here n = Physical EP # (continued)

Bit	Field	Description
12	Reserved	
11	RXTXFIFOEVTEN	RxTxFifoEvt DEPEVT event enable. IN EP: FIFO underrun enable. OUT EP: FIFO underrun enable • 0: event disabled • 1: event enabled
10	XFERNRDYEN	XferNotReady DEPEVT event enable • 0: event disabled • 1: event enabled
9	XFERINPROGEN	XferInProgress DEPEVT event enable • 0: event disabled • 1: event enabled
8	XFERCMPLEN	XferComplete DEPEVT event enable • 0: event disabled • 1: event enabled
7-5	Reserved	
4-0	INTRNUM	Interrupt/Event Buffer number on which interrupts for this EP are generated. Must be 0 if a single interrupter HW configuration.

Table 2-16. USB_DEPCMDPAR0 Fields for DEPCFG Command

Bit	Field	Description
31-30	Config Action	• 0 - Initialize EP state. This action is used when an endpoint is configured the first time. It will cause the data sequence number and flow control state to be reset. DEPCMDPAR2 will be ignored. The encoding of this action is backward compatible with software that previously set “Ignore Sequence Number” to 0. • 1 - Restore EP state. This action is used when reconfiguring an EP with saved state after hibernation. It will cause the data sequence number and flow control state to be restored from DEPCMDPAR2. • 2 - Modify EP state. This action is used when modifying an existing EP configuration, such as changing the DEPEVTEN event enable bits, interrupt number, or MaxPacketSize. The data sequence number and flow control state will be unchanged, and the DEPCMDPAR2 will be ignored. The encoding of this action is backward compatible with software that previously set “Ignore Sequence Number” to 1. • When issuing an Endpoint Configuration command with Config Action=Restore, parameter 2 must be filled in with the same value returned by the Get Endpoint State command prior to hibernation.
29-26	Reserved	
25-22	BRSTSIZ	Burst Size: • 0x0: MIN 1-beat burst • 0xF: MAX 16-beat burst
21-17	FIFONUM	FIFO number assigned to this EP. Control EP must have the same value for IN and OUT EP. This field should be set to 0 for all other OUT EPs. Device mode must use lower 16 TxFIFOs (if there are more than 16 TxFIFOs in dual-role mode).
16-14	Reserved	•
13-3	MPS	Maximum Packet Size, in bytes - applies to IN and OUT EPs. The application must program this field with the maximum packet size for the current USB EP. USB 3.0 supports up to 1024 bytes.
2:1	EPTYPE	Endpoint type - this is the transfer type supported by this SUB endpoint: • 0x0: CTRL Control EP • 0x1: ISOC Isochronous EP • 0x2: BULK Bulk EP • 0x3: INT Interrupt EP
0	Reserved	

2.5.2.2 Set EP Transfer Resource Configuration (DEPXFERCFG) Command

This command assigns a transfer resource to the endpoint. There must be only one transfer resource allocated per EP. Start Transfer causes the use of the transfer resource. End Transfer or an XferComplete event releases the transfer resource. If software attempts to allocate more transfer resources than have been configured in the hardware, this command will return an error in the CmdStatus/EventStatus field.

Parameter setting:

- USB_DEPCFG[n].CommandParam: unused
- Additional parameters: 32-bits stored in parameter 0.
  - USB_DEPCMDPAR0[n] - stored the number of transfer resources allocated to this EP.
  - USB_DEPCMDPAR1[n] - reserved
  - USB_DEPCMDPAR2[n] - reserved

Table 2-17. USB_DEPCMDPAR0 Fields for DEPXFERCFG Command

Bit	Field	Description
31-16	Reserved	
15-0	NUMXFERRES	Number of transfer resources allocated to this EP. Must be set to 1.

2.5.2.3 Get Data Sequence Number (DEPGETDSEQ) Command

For non-isochronous EP only. Returns the current data sequence number for the EP, allowing it to be saved, and later restored using the DEPCFG command.

Parameter settings:

- USB_DEPCFG[n].CommandParam: unused
- Additional parameters: None
  - USB_DEPCMDPAR0[n] - reserved
  - USB_DEPCMDPAR1[n] - reserved
  - USB_DEPCMDPAR2[n] - reserved

2.5.2.4 Set Stall (DEPSETSTALL) Command

Stalls all tokens to the EP, including if the EP is in NRDY state.

- OUT EP: ongoing packet completes normally, next OUT DP or token is stalled
- IN EP, SS: current burst completes normally, next ACK TP is stalled
- IN EP, HS/FS/LS: current packet completes normally, next IN token is stalled
- Control EP: only the OUT direction shall be stalled
- Isochronous EP: not applicable, do not use

Parameter settings:

- USB_DEPCFG[n].CommandParam: unused
- Additional parameters: None
  - USB_DEPCMDPAR0[n] - reserved
  - USB_DEPCMDPAR1[n] - reserved
  - USB_DEPCMDPAR2[n] - reserved

2.5.2.5 Clear Stall (DEPCSTALL) Command

Clears the previously applied stall for the EP.

- Control EP: stall cannot be cleared by software: Hardware clears the stalls automatically when the next SETUP token is received.
- Non-Control EP: stall must be cleared by software. This also resets the DSeqNum to 0.

- Isochronous EP: not applicable, do not use

Parameter settings:

- USB_DEPCFG[n].CommandParam: unused
- Additional parameters: None
  - USB_DEPCMDPAR0[n] - reserved
  - USB_DEPCMDPAR1[n] - reserved
  - USB_DEPCMDPAR2[n] - reserved

2.5.2.6 Start Transfer (DEPSTRTXFER) Command

Indicates that a Transfer Descriptor (TD), that is, a list of TRBs that comprise one or more transfers, is ready to be processed. A pointer to the TD is recorded and is kept until the transfer ends (XferComplete event or DEPENDXFER command).

Hardware assigns the transfer a resource index number (XferRscIdx), returned in the USB_DEPCMD[n] register and in the Command Complete event, for use in subsequent Update and End Transfer commands.

- Non-stream capable EP:
  - StreamID must be set to 0
  - HighPri is reserved
- Stream-capable EP:
  - StreamID must be non-zero and match the StreamID in the TRB at the Transfer Descriptor Address
  - HighPri may be set for one or more streams.
  - Commands is issued for multiple streams, until the resources in DEPXFERCFG.NumXferRes are used up

NOTE: Before issuing a start transfer command, software needs to verify whether the link is in U0. If the link is not in U0, software needs to bring the link to U0 by performing a remote wake-up. This is applicable to both SS and USB 2.0 speeds.
When a IN transfer is in progress, (i.e., started and not completed yet), and the software sees a link state change event to L1, then the software can use the GDBGFIFOSPACE register (write followed by a read) to determine if the TxFIFO is empty or not. If the TxFIFO is not empty, it can initiate a remote wake-up.

Parameter Settings:

- USB_DEPCFG.CommandParam:
  - For isoc EP: CommandParam = StartMicroFramNum: 16-bit Frame / micro frame number to which the first TRB of the transfer applies
  - For stream-capable EP: CommandParam = StreamID: 16-bit USB stream ID associated to this transfer.
- Additional parameters: 64-bit Transfer Descriptor Address, stored in:
  - USB_DEPCMDPAR0[n] - the upper 32 bits
  - USB_DEPCMDPAR1[n] - the lower 32 bits
  - USB_DEPCMDPAR2[n] - reserved

Table 2-18. USB_DEPCMDPAR1 Fields for DEPSTRTXFER Command

Bit	Field	Description
31-0	TDADDR_LOW	Transfer Descriptor Address, lower 32 bits. Since TRBs must be aligned on a 16-byte boundary, the 4 LSBs of this field must be 0.

Table 2-19. USB_DEPCMDPAR0 Fields for DEPSTRTXFER Command

Bit	Field	Description
31-0	TDADDR_HIGH	Transfer Descriptor Address, upper 32 bits. All 0 in 32-bit-address architectures.

2.5.2.7 Update Transfer (DEPUPDXFER) Command

In the case of a circular TRB buffer, causes the hardware to re-cache the TRB following a TRB update by the software. Update includes the setting of HWO bit back to 1. The indicated resource transfer index identifies the updated TRB. It must have been previously started, then have completed its transfer (XferComplete event or an DEPENDXFER command)

Special No Response Update Transfer command is issued by setting CmdAct=0 and CmdIOC=0. In this case, no command complete (EpCmdCmplt) event is generated, CmdAct never changes, and another command may be issued to the same EP immediately. Do not use when software depends on the XferNotReady event to set up TRBs.

Parameter settings:

- DEPCFG.CommandParam:
  - CommandParam[6:0] = USB_DEPCMD_i[22:16] = XferRscIdx. The 7-bit Transfer resource index, assigned by the hardware to the transfer when it is started (DEPSTRTXFER command).
- Additional parameters: None.
  - USB_DEPCMDPAR0[n] - reserved
  - USB_DEPCMDPAR1[n] - reserved
  - USB_DEPCMDPAR2[n] - reserved

2.5.2.8 End Transfer (DEPENDXFER) Command

Request DMA to stop for the EP/stream, because of an error condition. TRB is identified by its transfer resource index. ForceRM bit (USB_DEPCMD[n].HIPRI_FORCERM, the bit11) must be set to 1 to remove transfers from the queue. CmdIOC bit (USB_DEPCMD[n].CMDIOC, the bit8) must be set to 1 in order to an EP Command Complete event is generated after the transfer ends, to synchronize the conclusion of system bus traffic before the End Transfer command is completed. Effect of the command:

- IN EP: A packet being transmitted may be truncated with the DPPABORT ordered set, any pending ACKs from the USB are ignored.
- OUT EP: Any data in the receive FIFO is moved to the corresponding memory buffer, until a packet boundary.
- TD processing stops, TRB status is not updated.
- No XferComplete event is generated upon transfer end, only a CommandComplete event. Use the command in the followng cases:
- To close EP when handling USBReset or SetConfiguration
- After receiving a ClearFeature (STALL) control transfer, before issuing Clear Stall, then Start Transfer
- After a XferInProgress event when the TRB after the one that caused the XferInProgress event has HWO=0
- For isochronous EP, if the host stops moving data for many intervals, to wait for the host to restart

Parameter Settings:

- DEPCFG[n].CommandParam:
  - CommandParam.bit[6:0] = USB_DEPCMD[n].bit[22:16] = XferRscIdx: 7-bit Transfer resource index, assigned by the hardware to the transfer when it is started (DEPSTRTXFER command)
- Additional parameters: None.
  - USB_DEPCMDPAR0[n] - reserved
  - USB_DEPCMDPAR1[n] - reserved
  - USB_DEPCMDPAR2[n] - reserved

2.5.2.9 Start New Configuration (DEPSTARTCFG) Command

Command issued to start a new configuration, in the following cases:

- Only to EP 0 (USB_DEPCMD[0])
- After power-on-reset with XferRscIdx=0 before starting to configure EP 0 and 1. CmdIOC bit (USB_DEPCMD[n].CMDIOC, the bit8) must be set to 0 and CmdAct bit CmdIOC bit (USB_DEPCMD[n].CMDACT, the bit10) must be polled because EP 0 is not yet configured with a valid interrupt number at that point.
- With XferRscIdx=2 when it receives SetConfiguration before starting to configure EP > 1

Transfer resource allocation gets reset to the value in the XferRscIdx parameter (must be 0 or 2).

Parameter settings:

- DEPCFG.CommandParam:
  - CommandParam[6:0] = USB_DEPCMD[n].bit[22:16] = XferRscIdx. The 7-bit Transfer resource index, assigned by software when the new configuration is started (DEPSTARTCFG command)
- Additional parameters: None.
  - USB_DEPCMDPAR0[n] - reserved
  - USB_DEPCMDPAR1[n] - reserved
  - USB_DEPCMDPAR2[n] - reserved

2.5.3 Device Mode Events

When the device controller needs to notify the driver software of something that has happened on the USB, it generates an event. The controller writes the event to the next location in the appropriate event buffer depending on the type of event and the event buffer that was assigned to handle those events. Then the controller increments the GEVNTCOUNT register, and schedules an interrupt that may be delayed if interrupt moderation is enabled.

When the driver software receives the interrupt, it reads the GEVNTCOUNT register to see how many event records have been added to the event buffer. It only starts processing the event buffer entries if the GEVNTCOUNT is non-zero. (it is possible to receive an interrupt but GEVNTCOUNT is zero --> ignore the interrupt).

After software processes one or more events from the buffer, it must write the number of bytes that it processed, back to the GEVNTCOUNT register.

This subsection describes the events passed by the device controller (hardware) to the device controller driver (software) through the event buffer(s) mapped in system memory.

Events are classified into two broad categories: Device Events which pertain generally to the USB device as a whole (eg: USB reset, connect/disconnect events); and EndPoint Events which pertain to a specific endpoint (a transfer completed on my endpoint).

Most event records are composed of a single (32-bit) dword, except the Vendor device test LMP received event, which takes 3 dwords.

Table 2-20 gives a summary of all possible events.

- Events are decoded from right to left (LSBits to MSBits). The bit0 indicates:
  - 0 - EndPoint related events (EP events defined by DEPEVT)
  - 1 - Non-EndPoint events (Device events defined by DEVT)
- Each line is a different category of event, described in Section 2.5.3.1 for EP Events (DEPEVT), and Section 2.5.3.2 for Device events (DEVT). The number in the first column is a link to the corresponding event description table.
- Short field names (a1, a2, b1, etc.) are described in Table 2-24

Table 2-20. Device-Mode Events Summary

	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
1	EventParam																EventStatus						EP event type				Physical EP num (0..31)					0
2	StreamID (only for bulk EP that supports streams)																a1	b1	c1	d1			4'b1 = XferComplete				Physical EP num (0..31)					0
3	IsocMicroFrameNum (for debug)																a2	b1	c1	d1			4'b2 = XferInProgress				Physical EP num (0..31)					0
4	StreamID (only for bulk EP that supports streams)																a3		b3				4'b3 = XferNotReady				Physical EP num (0..31)					0
5	(unused)																(unused)						4'b4 = RxTxFifoEvt				Physical EP num (0..31)					0
6	StreamID (only for bulk EP that supports streams)																a5 (idem)						4'b6 = StreamEvt				Physical EP num (0..31)					0
7	EventParam[15:12]				Command Type				EventParam[7:0]								EventStatus						4'b7 = EpCmdCmplt				Physical EP num (0..31)					0
8	(unused)				4'b2 = DepXferCfg				(unused)								a7						4'b7 = EpCmdCmplt				Physical EP num (0..31)					0
9	(unused)				4'b3 = DepGetDSeq				(unused)			CurDatSeqNum					(unused)						4'b7 = EpCmdCmplt				Physical EP num (0..31)					0
10	(unused)				4'b6 = DepStrtXfer					XferRscIdx							a9						4'b7 = EpCmdCmplt				Physical EP num (0..31)					0
11									EvtInfo												Device event sub-type		non-EP event type									1
12									(unused)												4'b0 = DisconnEvt		7'b0 = Device-Specific event (DEVT)									1
13									(unused)												4'b1 = UsbRst		7'b0 = Device-Specific event (DEVT)									1
14									(unused)												4'b2 = ConnectDone		7'b0 = Device-Specific event (DEVT)									1
15									(unused)			a13									4'b3 = ULStChng		7'b0 = Device-Specific event (DEVT)									1
16									(unused)												4'b4 = WkUpEvt		7'b0 = Device-Specific event (DEVT)									1
17									(unused)												4'b6 = EOPF		7'b0 = Device-Specific event (DEVT)									1
18									(unused)												4'b7 = SOF		7'b0 = Device-Specific event (DEVT)									1
19									(unused)												4'b9 = ErrticErr		7'b0 = Device-Specific event (DEVT)									1
20									(unused)												4'b10 = CmdCmplt		7'b0 = Device-Specific event (DEVT)									1
21									(unused)												4'b11 = EvntOverflow		7'b0 = Device-Specific event (DEVT)									1
22									Vendor-Specific Device Test Field												4'b12 = VndrDevTstRcved		7'b0 = Device-Specific event (DEVT)									1
	Vendor-Defined Data (dword 1/2)																															
	Vendor-Defined Data (dword 2/2)																															

Table 2-21 lists the event types, as used in the event buffer (dword 0 bits [7:0])

Table 2-21. Event Type

bit [7:1]	bit [0]	Name	Comments
don't care	0	DEPEVT	Device-mode, Endpoint event
don't care	1	non-EP	Device-mode, Device event
0000000	1	DEVT	Device-mode, Device event

Table 2-22 lists the 4-bit EP events (DEPEVT), as used in the buffer (dword 0 bits [9:6]).

Table 2-22. Endpoint Event (DEPEVT) Types

DEPEVT	Name	Comments
0x0	reserved	
0x1	XferComplete	Transfer complete
0x2	XferInProgress	Transfer in progress
0x3	XferNotReady	Transfer not ready

Table 2-22. Endpoint Event (DEPEVT) Types (continued)

DEPEVT	Name	Comments
0x4	RxTxFifoEvt	Receive/Transmit FIFO event
0x5	reserved	
0x6	StreamEvt	Stream Event
0x7	EpCmdCmplt	Endpoint Command Complete
0x8 to 0xF	reserved	

Table 2-23 lists the 4-bit device-specific events (DEVT), as used in the buffer (dword 0 bits [11:8]).

Table 2-23. Device-Specific Event (DEVT) Types

DEVT	Name	Comments
0x0	DisconnEvt	Disconnect detected
0x1	UsbRst	USB reset detected
0x2	ConnectDone	Connect done event
0x3	ULStChng	USB or Link state change event.
0x4	WkUpEvt	Host resume (wakeup) detected
0x5	reserved	
0x6	EOPF	End of periodic frame
0x7	SOF	Start of (micro-)frame
0x8	reserved	
0x9	ErrticErr	Erratic error
0xA	CmdCmplt	Generic command complete
0xB	EvntOverflow	Event buffer overflow
0xC	VndrDevTstRcved	Vendor device test LMP received from link partner.
0xD to 0xF	reserved	

Table 2-24. Device-Mode Events Encoding, Short Fields

Field	Bits	Description
d1	12	XferComplete/XferInProgress EP event: 1'b1: Bus error occurred.
c1	13	XferComplete/XferInProgress EP event: 1'b1: TRB completed with short packet reception or last packet of isoc interval.
b1	14	XferComplete/XferInProgress EP event: 1'b1: IOC bit of the TRB that completed.
a1	15	XferComplete EP event: 1'b1: LST bit of the completed TRB.
a2	15	XferInProgress EP event: MissedIsoc: 1'b1: Interval did not complete successfully.
b3	13:12	XferNotReady EP event: requested stage when (control) transfer was not ready: 2'b00: Control SETUP request 2'b01: Control data request 2'b10: Control status request
a3	15	XferNotReady EP event: 1'b0: XferNotActive: Host initiated a transfer not present in the hardware. 1'b1: XferActive: Host initiated a transfer that is present but no valid TRBs are available.
a5	15:12	StreamEvt EP event (only for bulk EP that supports streams): 4'h2: StreamNotFound: No active, ready stream found in transfer resource cache. 4'h1: StreamFound: Active and ready stream found, traffic to the host initiated for that stream.

Table 2-24. Device-Mode Events Encoding, Short Fields (continued)

Field	Bits	Description
a7	15:12	EpCmdCmplt EP event, DepXferCfg command: 4'h1: Error: Software requesting more transfer resources than are configured in the hardware.
a9	15:12	EpCmdCmplt EP event, DepStrtXfer command:
b13	19:16	ULStChng device-specific event: Link state upon event. See DSTS register for encoding.
a13	20	ULStChng device-specific event: SS event. 1'b1: SS 1'b0: non-SS

2.5.3.1 Endpoint Events (DEPEVT)

Category of events is specific to data endpoints (EP). Several types of EP events exist, with different field structures.

Table 2-25. Endpoint Events (DEPEVT)

Description	
Endpoint event (DEPEVT)	
Size in event buffer:	1 dword
Key:	Event type = evt[0] = DEPEVT
Subfields:	Physical EP number = evt[5:1]
	EP event type = evt[9:6]
	EventStatus = evt[15:12]
	EventParam = evt[31:16]

Table 2-26. Transfer complete EP event (XferComplete)

Description	
Type of endpoint event generated upon completion (sucessful or not) of a IN or OUT USB transfer.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = DEPEVT
	EP event type = evt[9:6] = XferComplete
Other fields:	EventParam = Stream ID = (if applicable, i.e. BULK EP with stream support)
	EventStatus = Bus Error flag + last isoc / short packet flag + IOC of TRB + LST of TRB
	Physical EP number

Table 2-27. Transfer in progress EP event (XferInProgress)

Description	
Type of endpoint event generated during the processing of a still-ongoing transfer.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = DEPEVT
	EP event type = evt[9:6] = XferInProgress
Other fields:	EventParam = IsocMicroFrameNum
	EventStatus = Missed isoc flag + last isoc / short packet flag + IOC of TRB + LST of TRB
	Physical EP number (0 to 31)

Table 2-28. Transfer not ready EP event (XferNotReady)

Description	
Type of endpoint event that indicates receipt of a transaction when no TRBs are available for the EP. For isochronous IN endpoints, a zero-length packet is sent. For non-isochronous endpoints, an NRDY is sent.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = DEPEVT
	EP event type = evt[9:6] = XferNotReady
Other fields:	EventParam = Stream ID = (if applicable, i.e. BULK EP with stream support)
	EventStatus = Xfer active flag + control phase (status/data)
	Physical EP number (0 to 31)
Generation of the event:	For stream-capable EP, generated each time the host attempts a transaction.
	For non-stream-capable EP, generated only on the first transaction attempt after an EP command to this EP.
	For isochronous EP, generated only once prior to the Start Transfer command to communicate the current bus time.

Table 2-29. FIFO EP event (RxTxFIFOEvt)

Description	
Type of endpoint event generated upon FIFO underrun / overrun, for debug only.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = DEPEVT
	EP event type = evt[9:6] = RxTxFIFOEvt
Other fields:	Physical EP number (0 to 31)
Generation of the event:	Only possible when thresholding is enabled
	FIFO underrun for IN EP = USB transmission
	FIFO overrun for OUT, non-isochronous EP = USB reception

Table 2-30. Stream EP event (StreamEvt)

Description	
Type of EP event indicating that a stream-capable EP initiated a search within its transfer resource cache. The result of the search is in the EventStatus field.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = DEPEVT
	EP event type = evt[9:6] = StreamEvt
Other fields:	EventParam = Stream ID = (if applicable)
	EventStatus = Found / NotFound flag
	Physical EP number (0 to 31)

Table 2-31. Device EP command complete event (EpCmdCmplt)

Description	
Category of EP events generated upon completion of an EP command. Several sub-types exist depending on the command. Those with specific field structures are listed in the chapters below.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = DEPEVT
	EP event type = evt[9:6] = EpCmdCmplt
Other fields:	EventParam (contains the Command type, and more depending on the command)
	EventStatus (depending on the command)
	Physical EP number (0 to 31)

Table 2-32. Device EP command complete event (EpCmdCmplt): set transfer resource

Description	
Type of EP event generated upon completion of the Set transfer resource command	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = DEPEVT
	EP event type = evt[9:6] = EpCmdCmplt
	Command type = EventParam[11:8] = evt[27:24] = DepXferCfg
Other fields:	EventStatus = Error flag for the command
	Physical EP number (0 to 31)

Table 2-33. Device EP command complete event (EpCmdCmplt): get data sequence

Description	
Type of EP event generated upon completion of the Get data sequence command	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = DEPEVT
	EP event type = evt[9:6] = EpCmdCmplt
	Command type = EventParam[11:8] = evt[27:24] = DepGetDSeq
Other fields:	EventParam[4:0] = CurDatSeqNum : current data sequence number
	Physical EP number (0 to 31)

Table 2-34. Device EP command complete event (EpCmdCmplt): start transfer

Description	
Type of EP event generated upon completion of the Start transfer command	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = DEPEVT
	EP event type = evt[9:6] = EpCmdCmplt
	Command type = EventParam[11:8] = evt[27:24] = DepStartXfer
Other fields:	EventParam[7:0] = XferRscIdx : resource index number
	EventStatus = Error flag for the command
	Physical EP number (0 to 31)

2.5.3.2 Device events (DEVT)

All non-EP events are listed in Table 2-35 through Table 2-46. Note that the only type of non-EP event is the device event, that is, an event that applies to the entire device, as opposed to only an EP.

NOTE: DEVT described here must not to be confused with device-mode events, that is, all events generated by the controller when in USB device mode, including the EP events (DEPEVT) described in the previous section.

Table 2-35. Device events (DEVT)

Description	
Device event (DEVT)	
Size in event buffer:	1 or 3 dword depending on event
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
Other fields:	device event type = evt[11:8]
	EvtInfo = evt[23:16]

Table 2-36. Disconnect DEVT event (DisconnEvt)

Description	
Device event generated upon disconnect detection.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
	device event type = evt[11:8] = DisconnEvt
Other fields:	none (EvtInfo is unused)

Table 2-37. USB reset DEVT event (UsbRst)

Description	
Device event generated upon USB reset detection.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
	device event type = evt[11:8] = UsbRst
Other fields:	none (EvtInfo is unused)

Table 2-38. Connect done DEVT event (ConnectDone)

Description	
Device event generated upon USB connect detection.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
	device event type = evt[11:8] = ConnectDone
Other fields:	none (EvtInfo is unused)

Table 2-39. USB / link status change DEVT event (ULStChng)

Description	
Device event generated upon USB / Link status change. In superspeed, generated only when entering the following USB3.0 LTSSM states: Rx.Detect, SS.Disabled, Polling, U1, U2, U3, or Recovery (only from U1, U2, U3).	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
	device event type = evt[11:8] = WkUpEvt
Other fields:	EvtInfo[4] = SuperSpeed flag
	EvtInfo[3:0] = Link State

Table 2-40. Wakeup DEVT event (WkUpEvt)

Description	
Device event generated upon detection of USB resume from the remote host. WARNING: Event is not generated by the device's own remote wakeup, or by the resume normally transmitted by the host in reply to that.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
	device event type = evt[11:8] = WkUpEvt

Table 2-40. Wakeup DEVT event (WkUpEvt) (continued)

Description	
Other fields:	none (EvtInfo is unused)

Table 2-41. End of periodic frame DEVT event (EOPF)

Description	
Device event generated at the end of periodic frame.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
	device event type = evt[11:8] = EOPF
Other fields:	none (EvtInfo is unused)

Table 2-42. Start of frame DEVT event (SOF)

Description	
Device event generated at the start of (micro-)frame.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
	device event type = evt[11:8] = SOF
Other fields:	none (EvtInfo is unused)

Table 2-43. Erratic error DEVT event (ErrticErr)

Description	
Device event generated upon erratic error event: Rxvalid / rxactive signals incorrectly asserted (by the USB2.0 PHY) for more than 2 ms on UTMI interface. Device then goes to Suspended, with a ULStChng event generated. Recovery from that error requires a soft disconnect.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
	device event type = evt[11:8] = ErrticErr
Other fields:	none (EvtInfo is unused)

Table 2-44. Command Complete DEVT event (CmdCmplt)

Description	
Device event generated upon (non-EP) command completion.	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
	device event type = evt[11:8] = CmdCmplt
Other fields:	none (EvtInfo is unused)

Table 2-45. Event buffer overflow DEVT event (EvntOverflow)

Description
Device event generated upon event buffer overflow: One or more events may have been dropped after this event, when buffer became full.

Table 2-45. Event buffer overflow DEVT event (EvntOverflow) (continued)

Description	
Size in event buffer:	1 dword
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
	device event type = evt[11:8] = EvntOverflow
Other fields:	none (EvtInfo is unused)

Table 2-46. Vendor device test LMP received DEVT event (DevTstRcved)

Description	
Device event generated upon receipt of vendor device test LMP from link partner	
Size in event buffer:	3 dword
Key fields:	Event type = evt[0] = non-EP
	non-EP event type = evt[7:1] = DEVT
	device event type = evt[11:8] = DevTstRcved
Other fields:	EvtInfo = LMP's Vendor-Specific Device Test Field
	LMP's Vendor-defined data (2 dwords = 64 bits total)

2.6 USB Device (Peripheral) Programming Model

The DRD Controller in device mode uses a dedicated set of programming registers, prefixed with a D, and described in Section 3.2.9.

Operation in USB device mode is controlled using the following tools:

- Device registers - accessed by the software. Commands pass the address in main memory of the transfer request block (TRB) to the controller DMA.
- Device TRBs - mapped in main memory, TRBs describe the data buffers also in main memory. Initially created by the software driver, read and updated by the controller DMA.
- Data buffers - areas of main memory containing USB data to transmit or make available for received USB data. Data buffers are defined in the TRBs.
- Event buffer - mapped in main memory, the event buffer is a circular buffer containing USB events written by the device controller and read by the device software driver.
- Interrupt lines - flags core events

2.6.1 Register Initialization

2.6.1.1 Initialization after Power-On Reset or a System Reset

The DRD Core (Controller) contains global registers (prefixed by G) and device registers (prefixed by D) that are programmed to start operation and handler certain events.

When the device is powered on and the USB module is enabled from PSC control, it is recommended that software also perform a soft reset before initializing the following registers.

Table 2-47. Initialization on Power-On Reset or System Reset

Register	Description
DCTL	Reset the device core by setting DCTL.CSftRst=1, and waiting for a read to return 0.
GSBUSCFG0/1	
GTXTHRCFG / GRXTHRCFG	Enable thresholding if it is planned to use.
GCORID	Read the CORE ID for the core version information so that it can configure system accordingly.
GUID	Optionally, software can program the User ID into GUID register.

Table 2-47. Initialization on Power-On Reset or System Reset (continued)

Register	Description
GUSB2PHYCFG	Program following PHY configuration fields: USBTrdTim, FSIntf, PHYIf, and TOUTCal, according to the implementation.
GUSB2PIPECTL	Program the following PHY configuration fields: DatWidth, PrtOpDir according to the implementation.
GTXFIFOSIZn / GRXFIFOSIZn	Write these registers to allocate prefetch buffers for each Tx endpoint. Unless the packet sizes of the EPs are application-specific, it is recommended to use the default value.
GEVNTADRn / GEVNTSIZn / GEVNTCOUNTn	Depending on the number of interrupts allocated, program the Event Buffer Address and Size registers to point to the Event Buffer locations in system memory, the sizes of the buffers, and unmask the interrupt. Note: USB operation will stop if Event Buffer memory is insufficient, because the core will stop receiving/transmitting packets.
GCTL	Program this register to override ScaleDown, RAM clock selection, and clock gating parameters.
DCFG	Program device speed and periodic frame interval.
DEVTEN	At a minimum, enable USB Reset, Connection Done, and USB/Link State Change events.
DEPCMD0	Issue a DEPSTARTCFG command with DEPCMD0.XferRscIdx=0, CmdIOC=0 to initialize the transfer resource allocation. Poll CmdAct for completion.
DEPCMD0 / DEPCMD1	Issue DEPCFG command for physical EP0 and EP1 with the following characteristics, and poll CmdAct for completion: • USB EP number =0 or 1 (for physical EP0 and EP1). • FIFONum = 0 • XferNRdyEn and XferCmplEn = 1 • Maximum Packet Size = 512 • Burst Size = 0 • EPType=2'b00 (Control)
DEPCMD0 / DEPCMD1	Issue a DEPXFERCFG command for physical EP0 and EP1 with DEPCMDPAR0_0/1 set to 1, and poll CmdAct for completion
DEPCMD0	Prepare a buffer for a setup packet, initialize a setup TRB, and issue a DEPSTRTXFER command for physical EP0, pointing t the setup TRB. Poll CmdAct for completion Note: the core will attempt to fetch the setup TRB via the master interface after this command complete
DALEPENA	Enable physical EP0 and EP1 by writing 0x3 t this register
DCTL	Set DCTL RunStop=1 to allow the device to attach to the host. At this point, the device is ready to receive SOF packets, respond to control transfers on control EP0, and generate events

2.6.1.2 Initialization on USB Reset

To initialize the core as device, during USB Reset, the application must perform the following steps:

Table 2-48. Initialization on USB Reset

Register	Description
DEPCMD0	If a control transfer is still in progress, complete it and get the core into the "Setup a Control-Setup TRB / Start Transfer" state
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control EP0)
DEPCMDn	Issue a DEPCSTALL (ClearStall) command for any EP that was put into STALL mode prior to the USB Reset
DCFG	Set DevAddr to 0

2.6.1.3 Initialization on Connect Done

When this event is received, software must perform the following steps:

Table 2-49. Initialization on Connect Done

Register	Description
DSTS	Read this register to obtain the connection speed
GCTL	Program the RAMClkSel field to select the correct clock for the RAM clock domain. This field is reset to 0 after USB Reset, so it must be reprogrammed each time on Connect Done
DEPCMD0 / DEPCMD1	Issue a DEPCFG command (with Config Action set to Modify) for physical EP0 and EP1 using the same EP characteristics from Power-On Reset, but set MaxPacketSize to 512 (SS), 64 (HS), 8/16/32/64 (FS), or 8 (LS)
GUSB2CFG / GUSB3PIPECTL	Depending on the connected speed, write to the other PHY's control register to suspend it
GTXFIFOSIZn	Optionally, based on the new MaxPacketSize of IN EP0, software may choose to re-allocate the TX FIFO sizes by writing to these registers

2.6.1.4 Initialization on SetAddress Request

When the application receives a SetAddress request in SETUP packet, it performs the following steps:

Table 2-50. Initialization on SetAddress

Register	Description
DCFG	Program it with the device address received as part of the SetAddress request decoded from the SETUP packet.
DEPCMD1	After receiving the XferNotReady (Status) event, acknowledge the status stage by issuing a DEPSTRTXFER command pointing to a Status TRB. This step must be done after the DCFG register is programmed with the new device address
DEPCMD0 / DEPCMD1	Issue a DEPCFG command (with Config Action set to Modify) for physical EP0 and EP1 using the same EP characteristics from Power-On Reset, but set MaxPacketSize to 512 (SS), 64 (HS), 8/16/32/64 (FS), or 8 (LS)

At this point, the device is ready to receive micro-SOF/ITP and is configured to receive control transfers on control EP0 with a new address assigned.

2.6.1.5 Initialization on SetConfiguration or SetInterface Request

When the application receives a SetConfiguration or SetInterface request in a SETUP packet, it performs the following steps:

Table 2-51. Initialization on SetConfiguration or SetInterface Request

Register	Description
DALEPENA	Set this register to 0x3 to disable all EPs other than the default control EP0
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control EP0)
DEPCMD1	Issue a DEPCFG command (with Config Action field set to Modify) for physical EP1 using the current EP characteristics to re-initialize the TX FIFO allocation.
DEPCMD0	Issue a DEPSTRTCFG command with DEPCMD0.XferRscIdx set to 2 to re-initialize the transfer resource allocation
DEPCMDn	Issue a DEPCFG command (with Config Action field set to Initialize) for each EP that is present in the new configuration (except for the default control EP0) Note: control endpoints are bi-directional and must use consecutive even/odd physical EP numbers for the OUT/IN direction (such as 2/3, or 4/5), and the FIFONum must be configured to the same value in both directions
DEPCMDn	Issue a DEPXFERCFG command for each EP that is present in the new configuration (except for the default control EP0)
GTXFIFOSIZn	Optionally, based on the new configuration of IN EPs, software may choose to re-allocate the TX FIFO sizes by writing to these registers
DALEPENA	Enable the logical EPs that are active in the new configuration
DEPCMD1	After receiving the Xfer NotReady (Status) event, acknowledge the status stage by issuing a DEPSTRTXFER command pointing to a Status TRB. This step must be done after the previous steps to ensure the host does not access any other EPs that are being set up

Table 2-51. Initialization on SetConfiguration or SetInterface Request (continued)

Register	Description
GCTL	Program this register to override ScaleDown, RAM clock selection, and clock gating parameters.
DCFG	Program device speed and periodic frame interval.
DEVTEN	At a minimum, enable USB Reset, Connection Done, and USB/Link State Change events.
DEPCMD0	Issue a DEPSTARTCFG command with DEPCMD0.XferRscIdx=0, CmdIOC=0 to initialize the transfer resource allocation. Poll CmdAct for completion.
DEPCMD0 / DEPCMD1	Issue DEPCFG command for physical EP0 and EP1 with the following characteristics, and poll CmdAct for completion: • USB EP number =0 or 1 (for physical EP0 and EP1). • FIFONum = 0 • XferNRdyEn and XferCmplEn = 1 • Maximum Packet Size = 512 • Burst Size = 0 • EPType=2'b00 (Control)
DEPCMD0 / DEPCMD1	Issue a DEPXFERCFG command for physical EP0 and EP1 with DEPCMDPAR0_0/1 set to 1, and poll CmdAct for completion
DEPCMD0	Prepare a buffer for a setup packet, initialize a setup TRB, and issue a DEPSTRTXFER command for physical EP0, pointing t the setup TRB. Poll CmdAct for completion Note: the core will attempt to fetch the setup TRB via the master interface after this command complete
DALEPENA	Enable physical EP0 and EP1 by writing 0x3 t this register
DCTL	Set DCTL RunStop=1 to allow the device to attach to the host. At this point, the device is ready to receive SOF packets, respond to control transfers on control EP0, and generate events

At this point, the core is prepared to accept Start Transfer commands for the newly-configured EPs.

2.6.1.6 Initialization on Disconnect Event

When the application receives a Disconnect event, it must set DCTL[8:5]=5. Other than this, the core does not require any initialization. Because the DCTL.RunStop bit is still 1, the device attempts to reconnect to the host, at which time a USB Reset and Connect Done event will occur. However, if the application does not want to attempt to reconnect to the host, it should perform the steps in the next section.

NOTE: When DCFG.DevSpd is programmed for 2.0 only mode (such as, HS or FS), if the application wants to issue any commands to clear any pending transfers during a Disconnect interrupt, then it has to disable gusb2phycfg.SusPHY before issuing any commands and re-enable it after the commands have completed.

2.6.1.7 Device-Initiated Disconnect

If the application wants to disconnect from the host, it should perform the following the following actions:

Table 2-52. Initialization on Device-Initiated Disconnect

Register	Description
DEFCMD0	If a control transfer is still in progress, complete it and get the core into the "Setup a Control-Setup TRB / Start Transfer" state
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control EP0)
DCTL	Set DCTL.RunStop to 0 to disconnect from the host
DSTS	Poll DevCtrlHlt until it is 1

At this point, the device is disconnected from the host and will not attempt to reconnect.

2.6.1.8 Reconnect after Device-Initiated Disconnect

If the application decides it wants to reconnect to the host, it must follow the steps in Section 2.6.1.1.

2.6.2 Device Programming Model

The following sections describes the processes and data structures that the core uses to implement the USB 3.0 specifications.

2.6.2.1 USB and Physical Endpoints

Endpoints are referred to in two ways:

- As a USB endpoint number: USB endpoints are defined in the USB specification.
- As a physical endpoint resource number: the hardware has a fixed number of physical endpoint resources. Each resource is unidirectional and can be configured to refer to either direction of any USB endpoint.

The software always works on physical EPs and it knows how physical EP corresponds to USB EPs (logical). DALEPENA is the only register that has on enable bit per USB EP.

During SetConfiguration, software maps physical EP resources to the required USB EPs.

When a USB request comes in, the USB EP number gets converted to physical EP number in the core. Similarly, when a USB packet is sent out, the physical EP number is converted to the USB EP and sent out.

A USB control EP requires two physical EPs. One physical EP is mapped to the OUT direction of the Control EP, and the other one is mapped to the IN direction of the Control EP. Specifically the two physical EPs are used as the following:

- The setup stage of any control transfer uses the OUT direction physical EP.
- For a control write or 2-stage transfer, the Data stage (if present) uses the OUT direction physical EP and the Status stage uses the IN direction physical EP.
- For a control read transfer, the Data stage uses the IN direction physical EP and the Status stage uses the OUT direction physical EP.

2.6.2.2 Event Buffers

Hardware passes command completion, transfer progress, and asynchronous events to software through one or more Event Buffers.

To configure an Event Buffer, the software performs the following steps:

- Sets up an empty buffer in system memory.
- Writes the address of the buffer into GEVNTADRn. This address must be aligned to the event Buffer size.
- Writes the size of the buffer and interrupt mask into GEVNTSIZn. Depending on interrupt latency, enough Event Buffer space must be allocated to avoid lost interrupts or reduced performance.
- Write a 0 into GEVNTCOUNTn. This must the last step, as it enables the Event Buffer

After the Event Buffer has been configured, software must not change the size or address.

There is one interrupt line per Event Buffer that indicates there are one or more events present. Software reads one or more events out of the buffer and indicates to hardware how many events it processed by writing the byte count to the GEVNTCOUNTn register.

Clock crossing delays may result in the interrupt's continual assertion after software acknowledges the last event. Therefore, when the interrupt line is asserted, software must read the GEVNTCOUNT register and only process events if the GEVNTCOUNT is greater than 0.

Most events are 32 bits (see Table 2-20 for the encoding of each device events), and hardware writes event to Event Buffer one after another until buffer is full, then next event will be wrapped around to the first GEVNTADR value. In this way, the Event Buffer operates like a circular buffer with hardware writing to the "tail" of the buffer and software reading from the "head".

When an event occurs within the core, hardware checks the enable bit that corresponding to the event to decide whether the event will be written to the Event Buffer or not. The Event Buffer contains one of the following types of information:

- Endpoint-Specific Event (DEPEVT)
  - The DEPCFG endpoint-specific command specifies the enables and which Event Buffer to use for the these events.
- Device-Specific Event (DEVT)
  - The Generic Command Complete event is enabled through the DGCMD.CmdIOC field when the command is issued.
  - The rest of the device-specific events are enabled through the DEVTEN register
  - These events are written to the Event Buffer specified in the DCFG.IntrNum field.

The core always leaves one entry free in each Event Buffer. When the Event Buffer is almost full, hardware writes the Event Buffer Overflow event and the USB will eventually get stalled when EP start responding NRDY or the link layer will stop returning credits (in SuperSpeed). This event is an indication to software that it is not processing events quickly enough. During this time, events will be queued up internally. When software frees up Event BUffer space, the queued up events will be written out and the USB will return to normal operation.

2.6.2.3 Transfer and Buffer Rules

Buffers that are used to transfer data to and from an EP are defined using a Buffer Descriptor, which consists of one or more TRBs. ACHN flag in the TRB is used to identify the TRBs that comprise a Buffer Descriptor. Therefore, a Buffer Descriptor refers to a consecutive set of TRB data structures where the CHN flag is set in all TRBs, except the last. Note that a Buffer Descriptor may consist of a single TRB, whose CHN flag will not be set.

Software communicates the location of the first TRB by using the Start Transfer command on an EP. Even EPs that are not stream-capable use this command.

The core fetches TRBs from external memory starting at the address provided in the Start Transfer command parameters, continuing in a linear fashion and following the Link TRB until a TRB is encountered that has its LST bit set to 1 or HWO bit set to 0. Therefore, software must ensure that it has valid TRBs prepared before issuing the Start Transfer command to prevent the core from reading uninitialized memory.

Descriptor fetch requests are buffered within the hardware and handled separately from the progress of the current transfer. If software is immediately de-allocating the memory for TRBs based on the XferComplete event, it is recommended that software issue an End Transfer command, which flushes out any pipelined descriptor fetches, for the EP/transfer resource prior to the de-allocating the memory, to avoid a potential bus error.

While processing TRBs, two conditions may cause the core to write out an event and raise an interrupt line:

- TRB Complete
  - For OUT EPs, a packet is received which reduces the remaining byte count in the TRB buffer to zero.
  - For IN EPs, an acknowledgement is received for a transmitted packet which reduces the remaining byte count in the TRB buffer to zero.
- Short Packet Received:
  - For OUT EPs only. While writing to a TRB buffer, the EP receives a packet that is smaller than the EP's MaxPacketSize.

Table 2-53 describes the action taken by the controller when these conditions occur.

Table 2-53. Controller Actions Based on Device TRB Control Bits

Direction	TRB Complete	Short Packet	ISP	IOC	CHN	LST	CSP	Action
IN	Yes			X	0	1		XferComplete event
IN	Yes			0	X	0		No event
IN	Yes			1	X	0		XferInProgress event
OUT	Yes	No		X	0	1		XferComplete event
OUT	Yes	No		0	X	0		No event
OUT	Yes	No		1	X	0		XferInProgress event
OUT	X	Yes	X	X	X	X	0	XferComplete event Note: when a TRB whose CSP=0 and CHN-1 receives a short packet, the chained TRBs that follow it are not written back (e.g. the BUFSIZ and HWO fields remain the same as the software prepared value.
OUT	X	Yes	X	X	0	1	1	xferComplete event
OUT	X	Yes	X	X	1	0	1	Search for CHN=0, accumulate IOC and ISP, then follow CHN=0 rules Note: in the case of an OUT EP, if CHN=1 and CSP-1, and a short packet is received, the controller retires the TRB in progress and skip past the TRB where CHN=0, accumulating the ISP and IOC bits from each TRB. If ISP or IOC is set in any TRB, the core generates an XferInProgress event. HW does not set HWO=0 in skipped TRBs.
OUT	X	Yes	X	1	0	0	1	XferInProgress event
OUT	X	Yes	0	0	0	0	1	No event
OUT	X	Yes	1	0	0	0	1	XferInProgress event

On XferComplete and XferInProgress events, status bits in the event indicate LST, IOC, Short Packet Received, or Bus Error status. In the “fast-forward” case, the IOC status is accumulated from the skipped TRBs, not just the TRB that received the short packet.

When the hardware writes back the TRBs, it updates the BUFSIZ field to represent the remaining unused buffer.

2.6.2.4 Number of TRBs Rule

- Software must set up only one TRB for a control setup or status stage.
- If SW is preparing multiple transfers for an IN EP, it may be necessary to place a 0-length TRB between transfers that are MaxPacketSize aligned to indicate transfer boundaries to the host. This is not necessary, if the class driver and the host can handle bursting between transfers.
- If SW is preparing multiple transfers for an OUT EP, it needs to place a MaxPacketSize TRB between transfers, if it expects the host to transmit a 0-length packet between transfers.

2.6.2.5 TRB Control But Rules

Transfer control bits (see Section 2.5.1) must conform to the following restrictions:

- For OUT EPs, the CSP bit must be same in every TRB within a Buffer descriptor.
- Every time SW validates a TRBy by setting HWO=1, it must also issue an Update Transfer command to the core.
- SW sets the HWO=1 when it creates the TRB, and cannot modify it until HW resets it to 0. However,
  - SW must detect when a “fast-forward” occurs on a OUT EP that receives a short packet, since some TRBs in the chain may still have their HWO=1
  - HW will not clear HWO bit of a Link TRB. Therefore, SW can only modify a Link TRB if the TRB prior to the Link TRB has its HWO bit set to 0.
- The LST bit must not be set to 1 for isochronous endpoints.
- For a Setup or Status TRB, set CHN=0, LST=1, and CSP=0.

- For the data stage of a control transfer, set CSP=0 in all TRBs and LST=1 in the last TRB of the data stage.
- For Link TRBs, the LST, CHN, IOC, ISP, CSP, and Stream ID fields are ignored and must be set to 0.

2.6.2.6 Buffer Size Rules and Zero-Length Packets

The HW contains a cache that holds a fixed number of TRBs per transfer. For IN EPs, the following rules apply:

- The number of chained TRBs necessary to construct a single packet must never exceed this number minus 1. A maximum of one Link TRB can be present in the chain.
- If SW wants to indicate a transfer completion to the host by sending a zero-length packet after a multiple of MaxPacketSize, i must set up a zero-length TRB following the last TRB in the transfer.

For OUT EPs, the following rules apply:

- If the first TRB has CHN=1, its buffer size must be >= 1 byte.
- The total size of a Buffer Descriptor must be a multiple of MaxPacketSize.
- A received zero-length packet still requires a MaxPacketSize buffer. Therefore,if the expected amount of data to be received is a multiple of MaxPacketSize, software should add MaxPacketSize bytes to the buffer to sink a possible zero-length packet at the end of the transfer.

For IN and OUT EPs, the following rule applies:

- The BUFSIZ field in a Link TRB must be set to 0.

2.6.3 Transfer Setup Recommendations

SW can either set up transfers before the host attempts to move data on a EP (“preset” transfer) or can set up transfer on demand (“on-demand” transfer). When using preset transfers, SW can safely disable teh XferNotReady event in the EP configuration. However, when using on-demand transfers, the XferNotReady event must be enabled and Sw may not use the “No Response” variant of the Update Transfer command. The XferNotReady event is issued when the host attempts to move data on an EP when one of the following conditions is present:

- No previous transfer was started with Start Transfer.
- Not enough hardware-owned (HWO=1)TRBs are available to handle the requested data movement.

The XferNotReady event must not be disabled for control EPs because the event is an integral part of the control transfer handling.

Although there are many valid ways to set up transfers, it is recommended that user choose one of the three general mechanisms:

- When SW wants to set up one transfer at a time and has the entire buffer available for transfer, it must set up TRBs that point to the data buffers and in the last TRB it must set the LST bit and issue Start Transfer, which points to the first TRB location.
  - When the USB transfer complete, the core will notify the SW through the XferComplete event. The LST bit will be set in the status field of the event. The XferComplete will also release the transfer Resource.
  - A premature XferComplete event can happen before all the data buffers are exhausted, if there is a Bus Error or, in an OUT transfer a short packet has been received and CSP=0. During these conditions the TRB will have an updated BUFSIZ field which represents the amount of buffer remaining after the successful part of the data transfer.
  - SW can also set up an IOC bit TRB, so that it gets notified when data buffers are used up and can free them up sooner than waiting for the entire transfer to complete. One completing a TRB with IOC set, and not the last one, the core will issue an XferInProgress event with IOC set in the status field of the event.
- When SW wants to set up one transfer at a time, but it has fewer data buffers available than the full transfer size, it must set up circular TRBs (using a Link TRB), and also set up IOC bits in the TRBs. Depending on buffer allocation and interrupt frequency it can set IOC for once in “x” number of TRBs.
  - As soon as the USB transfer for a TRB is completed, and if IOC is set, the core will generate a n

XferInProgress event with IOC set in the status field.

- On seeing the event, SW reuses the TRB and updates it with the next data buffer. It also issues an Update Transfer command, indicating it has updated a TRB.
- If SW is slow and HD finds a TRB with HWO reset to 0, it waits for a Update Transfer command. Upon seeing the Update Transfer, it prefetches the TRB and continues the transfer.
- When SW reaches the end of the transfer, it sets the TRB LST bit. When HW completes the TRB, it issues an XferComplete event and releases the Transfer Resource.

- When the Device SW has multiple transfers to set up, it must set up circular TRBs and also set up CHN bits in all TRBs, except the last of the transfer. For OUT EPs, multiple transfers can be supported when the CSP field is et to 0 if each transfer has a multiple of MaxPacketSize bytes, since a short packet will end the transfer. If multiple transfers may contain short packets, the CSP field must be set to 1 to enable the next transfer to continue event if a short packet is received. Depending on buffer allocation and interrupt frequency, it can either set the IOC for once in “x” number of TRBs or in the last TRB of each transfer.
  - For OUT EPs, the transfer can finish prematurely due to a short packet from the host. In this case, the core processes the remaining TRBs, skipping the updates to these TRBs until it reaches TRB of the next transfer. While skipping these TRBs, if any of the TRB is with ISP=1 or IOC=1, it will generate the corresponding event once and ignore the interrupt settings of the remaining TRBs until a TRB of next transfer is reached.
  - Device SW can reclaim these skipped TRBs even though the HWO=1.
  - For IN EPs, SW cannot stop providing transfer while it is ending transfers with CHN=0 and LST=0, otherwise it is possible that the EP is left in a flow-controlled state on the USB.

2.7 USB Host Programming Model

The eXtensible Host Controller Interface for USB (see *eXtensible Host Controller Interface for USB (xHCI) revision 1.0 with errata (errata files 1.7)*) is a standardized interface between a USB host controller (HC) and system software (HCD, Host Controller Driver), based on a dedicated DMA and a standard control and status register set.

xHCI controls all USB speeds of the Host mode, that is, the SS through the SS PHY, as well as the HS, FS, and LS through the USB2 PHY. xHCI replaces and supersedes all previous host HCIs (HS-only EHCI, FS/LS OHCI and UHCI), and is therefore not backwards compatible with any of them.

The USB3SS’s USB Controller is fully compliant with xHC. Moreover, it contains some non-xHCI registers (global registers) which are implementation-specific and are required before the xHCI operation can start.

2.7.1 Global Registers Initialization

To initialize the core as Host, the application should perform the steps described in the xHCI Specification. Possibly, user might need to override some of global registers (non-xHCI registers) before it can start the HCD driver initialization, depends on their reset values:

Table 2-54. Override Global Registers in Host Configuration

Register	Description
GSBUSCFG0/1	
GTXTHRCFG / GRXTHRCFG	This is required only if you are planning to enable thresholding.
GCOREID	Software might need read this register for the Controller’s version and configure itself accordingly.
GUID	Optionally, the software can program this User ID register.
GUSB2PHYCFG	Program the following PHY configuration fields: USBTrdTIm, FSIntf, PHYIf, TOUTTCal, depends on whether the default values are satisfactory
GUSB3PIPECTL	program the following PHY configuration fields: DatWidth, PrtOpDir, depends on whether the default values are satisfactory
CGTXFIFOSIZn / GRXFIFOSIZn	Program these registers based on the speed used for the FIFO. Unless the packet sizes of the endpoints are application-specific, it is recommended to sue the default value
GCTL	Program this register to override the ScaleDown, RAM clock select, and clock gating parameters

Table 2-54. Override Global Registers in Host Configuration (continued)

Register	Description
GUCTL	If you want to improve the interoperability with different devices, program this register to override the behavior of the core in Host mode

2.7.2 Host Controller Capability Registers

The xHCI standard supports a number of options (so-called capabilities) and configurations (port count). Specific to the USB3SS, the main characteristics of it are:

- HCSPARAMS1.MAXPORTS = 2 - there are two xHCI ports (that is, one USB2.0 physical port + one USB3.0 physical port).
  - SUPTPRT2_DW2.CPC = 1 - one xHCI USB2.0-capable port (HS/FS/LS)
  - SUPTPRT2_DW2.CPO = 1 - the USB2.0-capable port is the xHCI port 1
  - SUPTPRT3_DW2.CPC = 1 - one USB3.0-capable port (super-speed)
  - SUPTPRT3_DW2.CPO = 2 - the USB3.0-capable port is the xHCI port 2
- HCSPARAMS1. AXINTRS = 16 - there are 16 event Interrupter (IRQ lines)

The current implementation is configured by default for xHCI version 1.0 mode. A legacy mode is also available, which implements xHCI version 0.96 (*eXtensible Host Controller Interface for USB (xHCI) revision 0.96*).

- Default mode: xHCI 1.0 (CAPLENGTH.HCIVERSION = 0x0100, bit[31:16])
- Legacy mode: xHCI 0.96 (CAPLENGTH.HCIVERSION = 0x0096)

NOTE: The following procedure must be used to switch the xHCI mode to the legacy version (xHCI 0.96).

The Host has the same extended capabilities regardless of the xHCI version, as described in the extended capability registers, as per the xHCI standard:

- Legacy support (not related to xHCI version)
- USB2.0 protocol support
- USB3.0 protocol support

Host Programming Model

2.7.3 xHCI Programming Model

For mode details on xHCI programming model, refer to the xHCI Specification.

2.8 USB OTG Programming Model

NOTE: Early devices in the Keystone II family did not support OTG operation, and this section will be updated accordingly after the OTG operation is supported on Keystone II devices.

2.9 Reset Considerations

At system level, POR reset, LPSC controlled module reset, and system warm reset are applicable to the USB3SS subsystem, and it doesn't support the reset isolation.

Internally, USB3SS provides register bits to reset only a portion of the subsystem.

2.9.1 Software Reset

Table 2-55 shows USB resets which are controlled by software.

Table 2-55. USB Software Resets

No.	Resets	Descriptions	Programming Sequence
1	Wrapper Reset	Reset all Wrapper registers to their default values	Write 1 to SYSCONFIG.SOFT_RESET (bit0) and wait until it is self-cleared
2	Core Reset	Reset all core registers and a portion of RAM0 to their default value.	• Set register bit FLADJ.SOFT_RESET=1 (bit31), to assert a software reset to the core. • Poll USB_USBSTS.CNR for 0 (bit11, Controller Not Ready), before accessing to any other core registers. The reset of the RAM-stored core registers can take several cycles to complete. • Clear FLADJ.SOFT_RESET=0, to de-assert the reset.
3	USB2 PHY Reset	Reset the USB2 PHY only	• Set GUSB2PHYCFG[31]=1 • Clear GUSB2PHYCFG[31]=0
4	USB3 PHY Reset	Reset the USB3 PHY only	• Set GUSB3PIPECFG[31]=1 • Clear GUSB3PIPECFG[31]=0
53	LPSC Module Reset	Reset the entire USB3SS subsystem.	See specific device data manual for LPSC assignment for USB3SS module, and the LPSC manual for how to issue a reset.

Moreover, the DCTL.CSftRst (bit30), the GCTL.CoreSoftReset (bit11), the USBCMD.HCRST (bit1), and the USBCMD.LHCRST (bit7) control the reset to a portion of the core too. See Appendix for details.

Following is a typical sequence of using GCTL.CoreSoftReset and the PHY resets to reset the core and PHY during software development:

- Set GUSB2PHYCFG[31]=1, GUSB3PIPECFG[31]=1, and GCTL.CoreSoftReset=1. This resets the PHYs and keeps the core in reset state.
- Clear GUSB2PHYCFG[31]=0 and GUSB3PIPECFG[31]=0 after they meet PHY reset duration (minum16 cycles of suspend clock). This removes the Reset to the PHYs.
- Wait for the PHY clock to stabilize and clear GCTL.CoreSoftReset=0. This ensures that the Reset to all the internal blocks are asserted when all the clocks are stable.

2.9.1.1 Active RAM Reset

The RAM contents are assumed unknown after reset. However, the RAM0 contains register values and data cache (from address 0 to USB3_DCACHE_DEPTH_INFO-1) that must be initialized. For that purpose, the core automatically writes all-zero words to a section of RAM0 immediately after coming out of the core reset

This sequential process takes time (over 1000 interconnect clock cycles), during which functional read and write to the RAM-mapped registers are stalled. To avoid the stalls, the Controller's status bit USB_USBSTS.CNR can be polled until cleared.

Although this is a standard Host mode status bit, the method also works to access RAM-mapped registers of the Device.

2.9.2 Hardware Reset Considerations

2.9.2.1 Power-On Reset

The device level power-on reset resets the entire device, including the USB3SS.

2.9.2.2 System Reset

The System Reset is a warm reset, it can be triggered by the system reset pins, by CCS, and by watchdog timer(s). The System Reset resets the entire USB3SS.

2.10 Initialization

The programming model sections (Section 2.6, and Section 2.7) have discussed the initialization steps of software, mainly focusing on the core registers related initialization process. Before those steps, some system level initialization should be done also, including:

- Program the USB_PHY_CTL4:
  - To enable Spread-spectrum, PHY_SSC_EN=1 (bit31).
  - To use external PHY reference clock, set REF_USB_PAD (bit30).
    - 1 - for K2K devices.
    - 0 - for K2E and K2L devices.
  - To enable the reference clock, set REF_SSP_EN=1 (bit29)
  - Select appropriate setting for the specific reference clock frequency (bit[27:22]).
  - Disable OTG if it is not used (bit15)
- Program the USB LPSC to release the USB3SS from reset, and enable it.
- Program the SYSCONFIG register to enable/disable clock gating to each of the clock domains in USB3SS according to the specific configuration.
- Optionally, issue controller reset before starting the software driver initialization.

2.11 Interrupt Support

2.11.1 Interrupt Events and Requests

The USB3SS has 18 interrupt outputs, they are:

- 16 interrupts associated with 16 programmable Event Rings
- One OABS interrupt that combines all OTG, ADP, and SER events
- One MISC interrupt for miscellaneous events, currently, it is associated with the USB DMA disable event only.

Table 2-56. USB Interrupt Outputs

Name of Interrupt	Descriptions
USB_INT00	USB Event Ring 0 interrupt
USB_INT01	USB Event Ring 1 interrupt
USB_INT02	USB Event Ring 2 interrupt
USB_INT03	USB Event Ring 3 interrupt
USB_INT04	USB Event Ring 4 interrupt
USB_INT05	USB Event Ring 5 interrupt
USB_INT06	USB Event Ring 6 interrupt
USB_INT07	USB Event Ring 7 interrupt
USB_INT08	USB Event Ring 8 interrupt
USB_INT09	USB Event Ring 9 interrupt
USB_INT10	USB Event Ring 10 interrupt
USB_INT11	USB Event Ring 11 interrupt
USB_INT12	USB Event Ring 12 interrupt
USB_INT13	USB Event Ring 13 interrupt
USB_INT14	USB Event Ring 14 interrupt
USB_INT15	USB Event Ring 15 interrupt
USB_OABSINT	USB OABS interrupt - combined all OTG events, ADP events, and SER (Soft error) events.
USB_MISCINT	USB Misc. interrupt.

NOTE: The mapping of the USB interrupts to hosts are device dependent. See the Data Manual of the specific device for the details.

2.11.2 Event Ring Interrupts

Event Ring is used by the controller to return command completion and asynchronous events to software. Each Event Ring is associated with an Interrupter, and the buffer of the ring is stored in system memory outside the USB3SS, and each Event Ring is associated with an Interrupter. An Interrupter asserts an interrupt if it is enabled and its associated Event Ring contains Event TRBs that require an interrupt.

KeyStone II devices support 16 interrupters. Consequently, it can support up to 16 Event Rings. The HCSPARAMS1.MaxIntrs determines the number of Interrupters.

For Host mode, each Interrupter consists of:

- an IMAN (interrupter management) register
- an IMOD (interrupter moderation) register
- an Event Ring defined by a set of registers: ERSTSZ[] / ERSTBA[] / ERDP[] (in Host mode)
- an Event Ring defined by a set of registers: GEVNTADR[] / GEVNTSIZ[] / GEVNTCOUNT[] (in Device mode).

Those even ring register set defines the address, the size, and an interrupt mask of an Event Ring. Software must initialize the address and the size at initialization time, and must not change their values after initialization.

2.11.2.1 Interrupt Moderation (Host Mode)

To reduce the number of interrupts that SW processes, an interrupt moderation scheme is implemented based on the IMOD register and the ERDP.EHB (Event Handler Busy) flag.

When an Interrupter is enabled (IMAN.IE=1) it begins looking for two conditions:

- IPE = 1 (Interrupt Pending Enable, an internal flag), and
- ERDP.EHB = 0 (Event Handler not busy)

If these two conditions are true, the followings will happen:

- IMAN.IP = 1 (Interrupt Pending)
- ERDP.EHB = 1 (Event Handler is busy)
- IMOD.IMODC is loaded with IMOD.IMODI

Moderation counter starts counting down, and another interrupt will not be asserted to the host until all following three conditions are met:

- IMOD.IMODC = 0 (the moderation counter has counted down to 0)
- IPE = 1 (the Interrupt Pending Enable flag is asserted), and
- ERDP.EHB = 0 (the Event Handler is not busy).

Then, IMOD.IMODC is reloaded with IMOD.IMODI, and the process repeats again.

Here, IPE is an Interrupter internal flag, and is managed as following:

- IPE is cleared to 0
  - When the Event Ring is initialized, or
  - If the Event Ring transitions to empty
- IPE is set to 1
  - When an Event TRB is inserted on the Event Ring and the TRB.BEI = 0

2.11.2.2 Event Processing

For Host mode event processing, refer to the xHCI Specification.

For Device mode, most events are 4 bytes, and register GEVNTCOUNT.EVNTCount holds the number of valid bytes in the Event Ring (at initialization, software sets it to 0)

Section 2.6.2.2 shows operations of the Event Buffer. Each time the controller writes a new event to the Event Buffer, the counter increases. Whenever the count is greater than zero, an interrupt will be generated (if enabled and may be delayed if moderated). Once the Event(s) has/have been processed, software needs to decrement the counter by the number of bytes that it processed.

To process pending events, software can either poll GEVNTCOUNT[x].EVNTCount for any pending events in a Event Ring, or using ISR for any interrupt enabled events. Following is a typical processing sequence of an ISR:

- Reads the associated Event Buffer registers to determine
  - Address of Event Buffer
  - Size of Event Buffer
  - Count of the number of valid bytes in the Event Buffer
- Reads 1 or more event TRBs from the Event Buffer, and processes them
- Decrements the GEVNTCOUNT[x].EVNTCount accordingly after processing
- Clears the IRQSTATUS register.
- Writes to the EOI register.

2.11.2.3 Primary Event Ring and Secondary Event Ring

The multiple Event Rings and interrupts are used to support multi-core systems. Each Event Ring will have its own set of registers as defined above. Event Ring 0 is the Primary Event Ring and the remaining Event Rings are Secondary Event Rings. The Primary Event Ring supports Command Completion Events, DMA Transfer Events, Host controller Events, Device controller Events, Port Status Change Events, and MFINDEX Wrap Events. The Secondary Event Rings support Transfer Events, Host Controller Events, and Device Controller Events.

The Primary Event Ring is required by the controller, but the 15 secondary Event Rings are optional - they are application dependents.

There is a one-to-one association between the 16 Event Rings and the 16 interrupts. The interrupts will only be active if the associated Event Ring is initialized for operation.

The Controller always leaves one entry free in each Event Ring. When the Event Ring is almost full, the controller writes the Event Buffer Overflow event and the USB will eventually stall when endpoints start responding NRDY or the link layer stops returning credits (in SS). This Event is an indication to software that it is not processing events quickly enough. During this time, the Controller will queue up the events internally. When software frees up Event Ring space, the queued events will be written to the appropriate Event Ring and the USB will return to normal operation.

2.11.2.4 Event Enable

There are several events that occur within the Controller. When the event occurs, the Controller checks the enable bit that corresponds with the event to determine if the event will be written to the Event Ring. The Controller also checks the interrupt number to determine which Event Ring to write the event to.

For the registers and bit fields associated with Events, Event Enables, and Event Ring selection, please refer to the xHCI standard specification document “eXtensible Host Controller Interface for USB (xHCI)” for additional information.

2.11.2.5 Device Endpoint Specific Events

For Device Endpoint Specific Events the DEPCMDPAR1(n) register contains the event enables and the associated interrupt number (and Event Ring) as shown in Table 2-57.

Table 2-57. Event Enables in DEPCMDPAR1[n] - DEPCFG Parameter-1 Register

Bit(s)	Description
13	Stream Event Enable
11	FIFO Underrun Enable (applies only to IN endpoints) FIFO Overrun Enable (applies only to OUT endpoints)
10	XferNotReady Enable
9	XferInProgress Enable
8	XferComplete Enable
4:0	Interrupt number

2.11.2.6 Device Specific Events

The Device Specific Events are controlled by registers: DGCMD, DEVTEN, and DCFG.

The Device Generic Command Register (DGCMD) enables software to send link management packets and notifications to specific devices. This register contains command, control, and status fields relevant to the current command. Bit8 is the Command Interrupt on Complete (CmdIOC) parameter. When this bit is set, the device controller issues a Generic Command Completion event after executing the command.

The Device Event Enable Register (DEVTEN) controls the generation of the Device Specific events. If an enable bit is set to 0, the event will not be generated.

Table 2-58. Device Event Enables in the DEVTEN Register

Bit(s)	Description
12	Vendor Device Test LMP Received Event Enable
11	Event Buffer Overflow Event Enable
10	Generic Command Completion Event Enable
9	Erratic Error Event Enable
7	Start of (micro) Frame Enable
6	End of Periodic Frame Event Enable
4	Resume / Remote Wake-up Detected Event Enable
3	USB / Link State Change Event Enable
2	Connection Done Enable
1	USB Reset Enable
0	Disconnect Detected Event Enable

The Device Configuration Register (DCFG) configures the controller in Device mode with a list of commands. Once the command has completed the Event will be sent to the Event Ring as defined in bits 16:12 Interrupt number.

2.11.3 OABS (OTG, ADP, BC, and SER) Interrupt

The OABS interrupt is asserted when an OTG, ADP, BC, and/or SER event is generated (and enabled). Table 2-59 lists the OABS Events and the associated type of event.

Table 2-59. OABS Events

Type	Name	Description
OTG	OTGConIDStsChngEvnt	Connecter ID Status Change Event
OTG	OTGADevBHostEndEvnt	A-device B-Host End Event
OTG	OTGADevHostEvnt	A-device host Event
OTG	OTGADevHNPChngEvnt	A-device HNP Change Event
OTG	OTGADevSRPDetEvnt	SRP Detect Event
OTG	OTGADevSessEndDetEvnt	Session End Detected Event
OTG	OTGBDevBHostEndEvnt	B-device B-Host End Event
OTG	OTGBDevHNPChngEvnt	B-device HNP Change Event
OTG	OTGBDevSessVldDetEvnt	Session Valid Detected Event
OTG	OTGBDevVBUSChngEvnt	VBUS Change Event
ADP	AdpPrbEvnt	ADP Probe Event
ADP	AdpSnsEvnt	ADP Sense Event
ADP	AdpTmoutEvnt	ADP Timeout Event
ADP	ADPRstCmpltEvnt	ADP Reset complete Event
BC	MV_ChngEvnt	Multi-Valued input changed Event
SER	SER_Evnt	RAM Soft Error Event

Table 2-60 shows the registers used to manage the OTG, ADP, and BC events.

Table 2-60. Registers Used to Manage OTG, ADP, and BC Events

Type	Events	Event Enable for Interrupt
OTG	OEVT	OEVTEN
ADP	ADPEVT	ADPEVTEN
BC	BEVT	BEVTEN

For SER event, the SER_CONTROL provides the similar management.

2.11.4 MISC Interrupt

The MISC interrupt is asserted when a bus transaction is generated by the USB controller after software has disabled the USB. The status of the USB is monitored by software by reading the various registers within the controller. Once software has determined that the USB is in a low power state, SS portion of the PHY is in the U3 state and the USB2 portion of the PHY is in suspend mode, software sets the SYSCONF.DMADISABLE parameter. Later, if an event occurs that requires the controller to generate a bus transaction, the DMADisableCLR event is generated, and the MISC interrupt is asserted (if enabled).

The SYSCONF.DMADISABLE parameter is automatically de-asserted when the bus transaction starts.

2.11.5 Interrupt Interface

The Wrapper module provides the interrupt interface to outside of the USB3SS. For each USB interrupt, the Wrapper provides following register set to manage the interrupt's Enable / Status / EOI (End Of Interrupt) operation:

- IRQ_EOI_name
- IRQ_STATUS_name[]
- IRQ_STATUS_RAW_name[]
- IRQ_ENABLE_SET_name[]
- IRQ_ENABLE_CLR_name[]

Here, the "_name" stands for:

- _MAIN for event ring interrupts.
- _OABS for OABS interrupt.
- _MISC for MISC interrupt.

The EOI function is to re-enable the detection of active interrupts. Writing to any EOI bits requires the controller to re-evaluate its pending sources and send another pulse.

The interrupt interface support both pulse and level output, controlled by HOST_HUT_CTRL.HOST_MSI_ENABLE.

2.11.6 Interrupt Multiplexing

Inside the USB3SS subsystem, the OABS interrupt is the combination of all OTG, ADP, BC, and SER events

2.12 DMA Event Support

Since the USB controller module contains a local DMA controller for data transfer between the system memory and its internal memory (FIFO/Cache/Queue, etc), there is no need for system level DMA.

2.13 Power Management

Power-saving features have been included in the USB module to allow it to be used in low power applications, and following power management features are available in the Controller core:

- Hibernation (save/restore) in Device (L1,L2,U3, Disconnected) and Host (U3, Disconnected)
- Hardware-controlled LPM in USB 2.0 Host
- Clock gating in Device (U1/U2/U3) and Host (U1/U2/U3)

The hardware-controlled LPM is a USB 2.0 Host-only feature where the host controller automatically detects the downstream port is idle for a certain amount of time and autonomously initiates an LPM token to the connected device. If the device accepts, the link is put into the standard USB 2.0 Suspend state.

The basic power management is the enabling and disabling of the clock domains. When one or more of the clocks are disabled, the contents of the registers as well as the contents of the memories are retained.

Table 2-61. Power Savings Support

USB 3.0 State	USB 2. 0 State	Device Power Savings	Host Power Savings
U1 (Hardware Initiated)	None	Core: clock gating. PHY: P1	Core: clock gating PHY: P1
U2 (Hardware Initiated)	LPM-L1 (Hardware Initiated)	Core: clock gating, LPM-L2 hibernation PHY: P2/Sleep	Core: clock gating. PHY: P2/Sleep
U3 (Software Initiated)	Suspend (Software Initiated)	Core: clock gating, hibernation PHY: P3/Suspend	Core: clock gating, hibernation PHY: P3/Suspend
Rx.Detect	None	None	Core: clock gating, hibernation PHY: P3/Suspend
SS.Disabled	None	Core: clock gating, hibernation PHY: P3/Suspend	None

The core implements clock gating in the following situations:

- In USB 2. 0 Device mode, when the UTMI suspend or l1_suspend signal is asserted and the core is idle
- In USB 3.0 Device mode, when the link is in U1/U2/U3 state and the core is idle
- In Host mode, the clock gating is enabled in the following situations: when the USB 2.0 port is in suspend or L1 Suspend state, and the USB 3.0 port is in U1/U2/U3 state and the core is idle.

Internal clock gating will apply to:

- Modules that use the RAM clock
- Some modules that use bus_clk

Internal clock gating will not apply to:

- Modules that use mac3_clk - when the USB 3.0 PHY is suspended, these modules switch to using susp_clk, so their power consumption is reduced but the clock is not completely stopped.
- Modules that use mac2_clk - when the USB 3.0 PHY is suspended, these modules switch to using susp_clk, so their power consumption is reduced but the clock is not completely stopped (note: the ADP logic is also clocked by the **susp_clk)**
- The module which uses bus_clk for detecting wakeup activity on the slave interface.

The internal clock gating can be enabled or disabled using the GCTL register with the default as enabled. Because the GCTL register is “sticky”, it will retain its value across soft reset and hibernation. The suspend clock’s frequency can ranges from 32 kHz to 125 MHz and needs to be defined in GCTL.PwrDnScale

For more details, see the power management sections in the standard specifications of USB 3.0, xHCI, USB 2.0, and USB2_LinkPowerManagement.

Other than those USB standard defined power management features, following power management features also are supported on the USB3SS.

2.13.1 PHY Power Management

The SS PHY has power states P0, P1, P2, and P3, corresponding to the SS LPM states of U0, U1, U2, and U3. In the P3 state, SS PHY does not drive the default functional clock, instead, the susp_clk is used in its place. Following are the registers for the PHY power management:

- GDBGLTSSM.PowerDown - shows power state of the PHY.
- GUSB3PIPECTL.request_p1p2p3 (bit24) - when set, the core always requests PHY power change from P0 to P1/P2/P3 during U0 to U1/U2/U3 transition.
- GUSB3PIPECTL.DelayP0TOP1P2P3 (bit18) - Delay PHY power change from P0 to P1/P2/P3 when link state changing from U0 to U1/U2/U3 respectively.

2.13.2 Disabling and Enabling Clocks via Software

Other than the clock gating scheme implemented inside the Controller core which is used for standard compliant power management of USB 3.0. The register SYSCONFIG also provides clock controls (enable/disable) to the multiple clock domains as shown Table 2-1, which is used while software knows which parts of the system will not be used. Following are typical scenarios for using those clocks.

Table 2-62. Scenarios for Clock Gating via the SYSCONFIG Register

Scenarios	bus_clk	pipe_clk	utmi_clk	susp_clk	ref_clk	phymmr_clk
All clocks on	On	On	On	On	On	On
SS mode only	On	On	Off	On	Off	On
UTMI mode only	On	Off	On	Off	On	On
PHY power down mode	Off	Off	Off	On	On	Off
All clocks off	Off	Off	Off	Off	Off	Off

2.13.3 Software Initiated Power Management

Following is the sequence of how software initiate a power management:

- Set GUSB3PIPECTL.bit17=1 (Suspend SS PHY) to suspend the SS PHY
- Set GUSB2PHYCFG.bit6=1 (Suspend USB2 PHY) to suspend the USB2 PHY
- Set GCTL.bit1=1 (Global hibernation) to force the core into hibernation
- Stop all USB activity by issuing Stop Endpoint Commands for each endpoint in the running state. This shall cause the Controller to update the respective Endpoint or Stream Context TR Dequeue Pointer and DCS fields.
- Set PORTSC1.bit[8:5]=5 (port link state) to put the SS port into RxDetect state
- Set PORTSC2.bit[8:5]=3 (port link state) to ut the UTMI port into U3 state
- Stop the Controller by setting USBCMD.bit0=0 (Run/Stop)

Once all the above steps have been completed, hardware can initiate a clock stop request.

2.14 Emulation Considerations

For debug purpose, the USB3SS Wrapper module provide a interface for monitoring internal signals of the USB core.

2.14.1 Debug Interface

This interface is for observing certain internal signals of the Controller module, consist of two registers:

- DEBUG_CFG - select the signals.
- DEBUG_DATA - observed values of the selected signals after being synchronized with the bus_clk.

Table 2-63 shows he signal mapping and selection for DEBUG_DATA.

Table 2-63. Signal Mapping and Selection of Debug Interface

Bit	DEBUG_CFG.SEL = 0x1	DEBUG_CFG.SEL = 0x2	DEBUG_CFG.SEL = 0x3
0	utmi_clk	pipe_txpclk_out	reserved
1	utmi_databus16_8	pipe_txdatak[0]	reserved
2	utmi_txvalid	pipe_txdatak[1]	reserved
3	utmi_txvalidh	pipe_txdatak[2]	reserved
4	utmi_txready	pipe_txdatak[3]	reserved
5	utmi_rxactive	pipe_elasticitybuffermode	reserved
6	utmi_rxvalid	pipe_txdetectrxloopback	reserved
7	utmi_rxvalidh	pipe_txelecidle	reserved

Table 2-63. Signal Mapping and Selection of Debug Interface (continued)

Bit	DEBUG_CFG.SEL = 0x1	DEBUG_CFG.SEL = 0x2	DEBUG_CFG.SEL = 0x3
8	utmi_rxerror	pipe_txoneszeros	reserved
9	utmi_reset	pipe_rxpolarity	reserved
10	utmi_l1_suspend_com_n	pipe_rxeqtraining	reserved
11	utmi_xcvrselect[0]	pipe_reset_n	reserved
12	utmi_xcvrselect[1]	pipe_powerdown[0]	reserved
13	utmi_termselect	pipe_powerdown[1]	reserved
14	utmi_opmode[0]	pipe_txdeemph[0]	reserved
15	utmi_opmode[1]	pipe_txdeemph[1]	reserved
16	utmi_linestate[0]	pipe_txmargin[0]	reserved
17	utmi_linestate[1]	pipe_txmargin[1]	reserved
18	utmi_idpullup	pipe_txmargin[2]	reserved
19	utmi_dppulldown	pipe_txswing	reserved
20	utmi_dmpulldown	pipe_rxtermination	reserved
21	utmi_drvvbus	pipe_rxpclk	reserved
22	utmi_chrgvbus	pipe_rxdatak[0]	reserved
23	utmi_dischrgvbus	pipe_rxdatak[1]	reserved
24	utmi_txbitstuffenable	pipe_rxdatak[2]	reserved
25	utmi_txbitstuffenableh	pipe_rxdatak[3]	reserved
26	utmi_hostdisconnect	pipe_rxvalid	reserved
27	utmi_iddig	pipe_phystatus	reserved
28	utmi_avalid	pipe_elecidle	reserved
29	utmi_bvalid	pipe_rxstatus[0]	reserved
30	utmi_vbusvalid	pipe_rxstatus[1]	reserved
31	utmi_sessend	pipe_rxstatus[2]	reserved

See the Appendix A for the encoding used for the LTSSM state (ltdb_link_state), the substate (ltdb_substate), and the u2_dssr_state.

USB Registers

Topic **Page**

3.1 USBSS Wrapper Registers

Table 3-1. USBSS Wrapper Registers

Offset	Acronym	Register Description	Section
0x0000	REVISION	Revision identifier	Section 3.1.1
0x0010	SYSCONFIG	System Control register controls various parameters of the master and slave interfaces	Section 3.1.2
0x0018	IRQ_EOI_MAIN	End-Of-Interrupt register flags the end of current event to allow the generation of further pulses on the MAIN event line when pulsed output is used.	Section 3.1.3
0x0020 + 0x10 * N	IRQ_STATUS_RAW_MAIN[N]	Raw Status register of the MAIN core interrupt. Also writing a 1 to a specific bit allows user to manually trigger an interrupt event.	Section 3.1.4
0x0024 + 0x10 * N	IRQ_STATUS_MAIN[N]	“Regular” Status of the MAIN core interrupt register. It is set only when it has been enabled. Also, writing a 1 to a specific bit allows user to manually clear a pending interrupt source.	Section 3.1.5
0x0028 + 0x10 * N	IRQ_ENABLE_SET_MAIN[N]	Set the ENABLE control of the MAIN core interrupt line.	Section 3.1.6
0x002C + 0x10 * N	IRQ_ENABLE_CLR_MAIN[N]	Clear the ENABLE control of the MAIN core interrupt line.	Section 3.1.7
0x042C	IRQ_EOI_MISC	End-Of-Interrupt register allows the CPU to acknowledge completion of an interrupt by writing to the EOI for MISC interrupt sources.	Section 3.1.8
0x0430	IRQ_STATUS_RAW_MISC	Raw Status of the miscellaneous interrupt. Writing a 1 to a specific allows user to manually trigger an interrupt event.	Section 3.1.9
0x0434	IRQ_STATUS_MISC	“Regular” Status of the miscellaneous interrupt line, set only when enabled. Writing a 1 to a specific bit allows user to manually clear a pending interrupt source.	Section 3.1.10
0x0438	IRQ_ENABLE_SET_MISC	Set the ENABLE control of the miscellaneous interrupt line.	Section 3.1.11
0x043C	IRQ_ENABLE_CLR_MISC	Clear the ENABLE control of the miscellaneous interrupt line.	Section 3.1.12
0x044C	IRQ_EOI_OABS	End-Of-Interrupt. Flag the end of current event to allow the generation of further pulses on the OABS event line when pulsed output is used.	Section 3.1.13
0x0450	IRQ_STATUS_RAW_OABS	Raw Status of the OABS interrupt line. Writing a 1 to a specific allows user to manually trigger an interrupt event.	Section 3.1.14
0x0454	IRQ_STATUS_OABS	“Regular” Status of the OABS interrupt line, set only when enabled. This is IRQ_STATUS_RAW_MAIN after being enabled. Writing a 1 to a specific bit allows user to manually clear a pending interrupt source.	Section 3.1.15
0x0458	IRQ_ENABLE_SET_OABS	Set the ENABLE control of the OABS interrupt line.	Section 3.1.16
0x045C	IRQ_ENABLE_CLR_OABS	Clear the ENABLE control of the OABS interrupt line.	Section 3.1.17
0x0508	TXFIFO_DEPTH	Tx FIFO Depth register indicates the size of the Tx FIFO in 64-bit words in the RAM1.	Section 3.1.18
0x050C	RXFIFO_DEPTH	Rx FIFO Depth register indicates the size of the Rx FIFO in 64-bit words in the RAM2	Section 3.1.19
0x0600	SER_CONTROL	Soft Error Rate control register for enabling specific error correction and /or detection functions.	Section 3.1.20
0x0604	SER_STATUS	Soft Error Rate status register for diagnosing the SER event.	Section 3.1.21
0x0608	SER_ADDRESS	Soft Error Rate address register records the address of the RAM that had the error event.	Section 3.1.22
0x0614	USBCONFIG	USB Configuration register	Section 3.1.23
0x0704	FLADJ	Register for Jitter adjustment as well as software reset to the xHC controller	Section 3.1.24

Table 3-1. USBSS Wrapper Registers (continued)

Offset	Acronym	Register Description	Section
0x0708	DEBUG_CFG	Debug Configuration register controls the selection of signals to be polled via the debug interface.	Section 3.1.25
0x070C	DEBUG_DATA	Debug Data register records the polled result of the signals selected by the DBUG_CFG register.	Section 3.1.26
0x0710	DEV_EBC_EN	Device mode External Buffer Control Enable register. Used on devices which support Trace over USB feature, such as K2E and K2L.	Section 3.1.27
0x0714	HOST_HUB_CTRL	Host or Hub Control register - a collection of various input signals that control the xHC controller's Host or Hub interfaces.	Section 3.1.28

NOTE: The base address of the above registers is device dependent. The user needs to refer to the specific device's data sheet for it. As an example, on K2K device, it is 0x02680000.

3.1.1 REVISION [Offset = 0x0000]

IP Revision Identifier (X.Y.R)

Used by software to track features, bugs, and compatibility

Figure 3-1. REVISION Register

31 30	29 28	27 16	15 11	10 8	7 6	5 0
SCHEME	BU	FUNC	R_RTL	X_MAJOR	CUSTOM	Y_MINOR
R--0x1	R-0	R-0x791	R-0x8	R-0x3	R-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-2. REVISION Register Field Descriptions

Bit	Name	Description
31-30	SCHEME	Old vs. current REVISION encoding scheme.
29-28	BU	Business Unit.
27-16	FUNC	Function indicates a software compatible module family. If there is no level of software compatibility a new Func number (and hence REVISION) should be assigned.
15-11	R_RTL	RTL Version (R), maintained by IP design owner.
10-8	X_MAJOR	Major Revision (X).
7-6	CUSTOM	Indicates a special version for a particular device. Consequence of use may avoid use of standard Chip Support Library (CSL) / Drivers. • STANDARD (0) = Non custom (standard) revision
5-0	Y_MINOR	Minor Revision (Y), maintained by IP specification owner.

3.1.2 SYSCONFIG [Offset = 0x0010]

Controls various parameters of the master and slave interfaces.

Figure 3-2. SYSCONFIG Register

31 17	16	15	14 13
Reserved	DMADISABLE	Reserved	REFCLKEN_N1	SUSPCLKEN_N
R-0	R/W-0	R-0	R/W-0	R/W-0

12	11	10	9	8	7 1	0
PIPECLKEN_N	Reserved	UTMICLKEN_N	PHYMMRCLKEN_N	BUSCLKEN_N	Reserved	SOFT_RESET
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

Legend: R = Read only; W = Write only; - *n* = value after reset

Table 3-3. SYSCONFIG Register Field Descriptions

Bit	Name	Description
31-17	Reserved	
16	DMADISABLE	Disable/Enable control of the DMA master (initiator) to block read/write accesses. Bit is auto-cleared (to 0) by HW in case of outgoing access, but must be set (to 1) manually. • DISABLED (1) = DMA is disabled, outgoing read/write accesses are blocked. When in smart-standby mode, standby is requested. • ENABLED (0) = DMA is enabled, outgoing read/write accesses are possible. When in smart-standby mode, standby exit is requested. • DISABLE (1) = Disable the DMA. SW must ensure that there are no ongoing transactions before setting this bit. The disabling can only be done by software. • ENABLE (0) = Enable the DMA. The enabling can also be done by the hardware, i.e. bit it auto-cleared.
15	Reserved	
14	REFCLKEN_N1	Active low clock enable for ref_clk • 0 = enable • 1 = disable
13	SUSPCLKEN_N	Active low clock enable for suspend_clk • 0 = enable • 1 = disable
12	PIPECLKEN_N	Active low clock enable for pipe_clk • 0 = enable • 1 = disable
11	Reserved	
10	UTMICLKEN_N	Active low clock enable for utmi_clk • 0 = enable • 1 = disable
9	PHYMMRCLKEN_N	Active low clock enable for phymmr_clk • 0 = enable • 1 = disable
8	BUSCLKEN_N	Active low clock enable for bus_clk. The only way recover from disabling the bus_clk is to reset the USB. • 0 = enable • 1 = disable
7-1	Reserved	
0	SOFT_RESET	Software reset of the USB3SS module. This bit will automatically clear inself after several cycles. • Write 0 = No action. • Write 1= Initiate software reset. • Read 0 = No action • Read 1 = Reset ongoing

3.1.3 IRQ_EOI_MAIN [Offset = 0x0018

Software End-Of-Interrupt, MAIN lines: Flag the end current event to allow the generation of further pulses on the line.

Allows the CPU to acknowledge completion of an interrupt by writing 1 to the bit(s) of the interrupt source(s) of the EOI register. An eoi_write signal will be generated and another interrupt will be triggered if interrupt source(s) remain. This register will be reset one cycle after it has been written to.

Unused when using the level interrupt line (depending on module integration).

Figure 3-3. IRQ_EOI_MAIN Register

31	16	15	14	13	12	11	10
Reserved		EOI15	EOI14	EOI13	EOI12	EOI11	EOI10
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

9	8	7	6	5	4	3	2	1	0
EOI9	EOI8	EOI7	EOI6	EOI5	EOI4	EOI3	EOI2	EOI1	EOI0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-4. IRQ_EOI_MAIN Register Field Descriptions

Bit	Name	Description
31-16	Reserved	
15	EOI15	Write 1 to flag End Of Interrupt "main" #15 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
14	EOI14	Write 1 to flag End Of Interrupt "main" #14 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
13	EOI13	Write 1 to flag End Of Interrupt "main" #13 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
12	EOI12	Write 1 to flag End Of Interrupt "main" #12 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
11	EOI11	Write 1 to flag End Of Interrupt "main" #11 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
10	EOI10	Write 1 to flag End Of Interrupt "main" #10 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
9	EOI9	Write 1 to flag End Of Interrupt "main" #9 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
8	EOI8	Write 1 to flag End Of Interrupt "main" #8 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
7	EOI7	Write 1 to flag End Of Interrupt "main" #7 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
6	EOI6	Write 1 to flag End Of Interrupt "main" #6 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action

Table 3-4. IRQ_EOI_MAIN Register Field Descriptions (continued)

Bit	Name	Description
5	EOI5	Write 1 to flag End Of Interrupt "main" #5 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
4	EOI4	Write 1 to flag End Of Interrupt "main" #4 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
3	EOI3	Write 1 to flag End Of Interrupt "main" #3 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
2	EOI2	Write 1 to flag End Of Interrupt "main" #2 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
1	EOI1	Write 1 to flag End Of Interrupt "main" #1 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action
0	EOI0	Write 1 to flag End Of Interrupt "main" #0 • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action

3.1.4 IRQ_STATUS_RAW_MAIN[N] [Offset = 0x0020+0x10*N, N=0-15]

Figure 3-4. IRQ_STATUS_RAW_MAIN Register

31	1	0
Reserved		IRQ_STATUS_RAW_MAIN
R-0		R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-5. IRQ_STATUS_RAW_MAIN Register Field Descriptions

Bit	Name	Description
31-1	Reserved	
0	IRQ_STATUS_RAW_MAIN	Write 0: No action Write 1: Set event Read 0: No event pending Read 1: Event pending

3.1.5 IRQ_STATUS_MAIN[N] [Offset = 0x0024+0x10*N, N=0-15]

Figure 3-5. IRQ_STATUS_MAIN Register

31	1	0
Reserved		IRQ_STATUS_MAIN
R-0		R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-6. IRQ_STATUS_MAIN Register Field Descriptions

Bit	Name	Description
31-1	Reserved	
0	IRQ_STATUS_MAIN	Write 0: No action Write 1: Clear event Read 0: No event pending Read 1: Event pending

3.1.6 IRQ_ENABLE_SET_MAIN[N] [Offset = 0x0028+0x10*N, N=0-15]

Figure 3-6. IRQ_ENABLE_SET_MAIN Register

31	0
COREIRQ_EN	
R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-7. IRQ_ENABLE_SET_MAIN Register Field Descriptions

Bit	Name	Description
31-0	COREIRQ_EN	IRQ enable for main core interrupt. • Write 0: No action • Write 1: Enable event • Read 0: Event is disabled • Read 1: Event is enabled

3.1.7 IRQ_ENABLE_CLR_MAIN[N] [Offset = 0x002C+0x10*N, N=0-15]

Figure 3-7. IRQ_ENABLE_CLR_MAIN Register

31	0
COREIRQ_EN	
R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-8. IRQ_ENABLE_CLR_MAIN Register Field Descriptions

Bit	Name	Description
31-0	COREIRQ_EN	IRQ enable for main core interrupt. • Write 0: No action • Write 1: Disable event (clear the Enable) • Read 0: Event is disabled • Read 1: Event is enabled

3.1.8 IRQ_EOI_MISC [Offset = 0x042C]

Software End-Of-Interrupt, MISC line: Flag the end current event to allow the generation of further pulses on the line.

Unused when using the level interrupt line (depending on module integration).

Figure 3-8. IRQ_EOI_MISC Register

31	1	0
Reserved		EOI_MISC
R-0		R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-9. IRQ_EOI_MISC Register Field Descriptions

Bit	Name	Description
31-1	Reserved	
0	EOI_MISC	Write 1 to flag End Of Interrupt "MISC" • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action

3.1.9 IRQ_STATUS_RAW_MISC [Offset = 0x0430]

The IRQ_STATUS_RAW_MISC register allows the USBSS interrupt sources to be manually set when writing a 1 to a specific bit.

Figure 3-9. IRQ_STATUS_RAW_MISC Register

31	18	17	16	0
Reserved		DMADISABLECLR	Reserved	
R-0		R/W-0	R-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-10. IRQ_STATUS_RAW_MISC Register Field Descriptions

Bit	Name	Description
31-18	Reserved	
17	DMADISABLECLR	Set when SYSCONFIG.DMADISABLE=1 and HW generates an AXI transaction
16-0	Reserved	

3.1.10 IRQ_STATUS_MISC [Offset = 0x0434]

The IRQ_STATUS_MISC register allows the USBSS interrupt sources to be manually cleared when writing a 1 to a specific bit.

Figure 3-10. IRQ_STATUS_MISC Register

31	18	17	16	0
Reserved		DMADISABLECLR	Reserved	
R-0		R/W-0	R-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-11. IRQ_STATUS_MISC Register Field Descriptions

Bit	Name	Description
31-18	Reserved	
17	DMADISABLECLR	Set when SYSCONFIG.DMADISABLE=1 and HW generates an AXI transaction
16-0	Reserved	

3.1.11 IRQ_ENABLE_SET_MISC [Offset = 0x0438]

The IRQ_ENABLE_SET_MISC register allows the USBSS interrupt sources to be manually enabled when writing a 1 to a specific bit.

Figure 3-11. IRQ_ENABLE_SET_MISC Register

31	18	17	16	0
Reserved		DMADISABLECLR	Reserved	
R-0		R/W-0	R-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-12. IRQ_ENABLE_SET_MISC Register Field Descriptions

Bit	Name	Description
31-18	Reserved	
17	DMADISABLECLR	Event enable
16-0	Reserved	

3.1.12 IRQ_ENABLE_CLR_MISC [Offset = 0x043C]

The IRQ_ENABLE_CLR_MISC register allows the USBSS interrupt sources to be manually disabled when writing a 1 to a specific bit.

Figure 3-12. IRQ_ENABLE_CLR_MISC Register

31	18	17	16	0
Reserved		DMADISABLECLR	Reserved	
R-0		R/W-0	R-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-13. IRQ_ENABLE_CLR_MISC Register Field Descriptions

Bit	Name	Description
31-18	Reserved	
17	DMADISABLECLR	Event enable
16-0	Reserved	

3.1.13 IRQ_EOI_OABS [Offset = 0x044C]

Software End-Of-Interrupt, OABS line: Flag the end current event to allow the generation of further pulses on the line.

Unused when using the level interrupt line (depending on module integration).

Figure 3-13. IRQ_EOI_OABS Register

31	1	0
Reserved		EOI_OABS
R-0		R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-14. IRQ_EOI_OABS Register Field Descriptions

Bit	Name	Description
31-1	Reserved	
0	EOI_OABS	Write 1 to flag End Of Interrupt "OABS" • READ0 (0) = Read returns zero • EOI (1) = Flag EOI • NOOP (0) = No action

3.1.14 IRQ_STATUS_RAW_OABS [Offset = 0x0450]

The IRQ_STATUS_RAW_OABS register allows the OTG, ADP, BC (not on K2 devices), and SER interrupt sources to be manually triggered when writing a 1 to a specific bit. A read of this register returns the interrupt event pending value. General actions per bit:

- Write 0: No action
- Write 1: Set event
- Read 0: No event pending
- Read 1: Event pending

Figure 3-14. IRQ_STATUS_RAW_OABS Register

31	28	27	26	25
Reserved		OTGXHCIRUNSTPSETEVNT	OTGDEVRUNSTPSETEVNT	OTGHIBENTRYEVNT
R-0		R/W-0	R/W-0	R/W-0
24		23	22	21
OTGCONIDSTSCHNGEVNT		HRRCONFNOTIFEVNT	HRRINITNOTIFEVNT	OTGADEVIDLEEVNT
R/W-0		R/W-0	R/W-0	R/W-0
20		19	18	17
OTGADEVBHOSTENDEVNT		OTGADEVHOSTEVNT	OTGADEVHNPCHNGEVNT	OTGADEVSRPDETEVNT
R/W-0		R/W-0	R/W-0	R/W-0
16		15	12	11
OTGADEVSESSENDDETEVNT		Reserved	OTGBDEVBHOSTENDEVNT	
R/W-0		R-0	R/W-0	
10		9	8	7
OTGBDEVHNPCHNGEVNT		OTGBDEVSESSVLDDETEVNT	OTGBDEVVBUSCHNGEVNT	ADPPRBEVNT
R/W-0		R/W-0	R/W-0	R/W-0

6	5	4	3	2	1	0
ADPSNSEVNT	ADPTMOUTEVNT	ADPRSTCMPLTEVNT	Reserved	Reserved	Reserved	SER_EVENT
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-15. IRQ_STATUS_RAW_OABS Register Field Descriptions

Bit	Name	Description
31-28	Reserved	
27	OTGXHCIRUNSTPSETEVNT	Interrupt status for OTGXhciRunStpSetEvnt
26	OTGDEVRUNSTPSETEVNT	Interrupt status for OTGDevRunStpSetEvnt
25	OTGHIBENTRYEVNT	Interrupt status for OTGHibEntryEvnt
24	OTGCONIDSTSCHNGEVNT	Interrupt status for OTGConIDStsChngEvnt
23	HRRCONFNOTIFEVNT	Interrupt status for HRRConfNotifEvnt
22	HRRINITNOTIFEVNT	Interrupt status for HRRInitNotifEvnt
21	OTGADEVIDLEEVNT	Interrupt status for OTGADevIdleEvnt
20	OTGADEVBHOSTENDEVNT	Interrupt status for OTGADevHNPChngEvnt
19	OTGADEVHOSTEVNT	Interrupt status for OTGADevHNPChngEvnt
18	OTGADEVHNPCHNGEVNT	Interrupt status for OTGADevHNPChngEvnt
17	OTGADEVSRPDETEVNT	Interrupt status for OTGADevSRPDetEvnt
16	OTGADEVSESSENDDETEVNT	Interrupt status for OTGBDevBHostEndEvnt
15-12	Reserved	
11	OTGBDEVBHOSTENDEVNT	Interrupt status for OTGBDevBHostEndEvnt
10	OTGBDEVHNPCHNGEVNT	Interrupt status for OTGBDevHNPChngEvnt
9	OTGBDEVSESSVLDDETEVNT	Interrupt status for OTGBDevSessVldDetEvnt
8	OTGBDEVVBUSCHNGEVNT	Interrupt status for OTGBDevVBUSChngEvnt
7	ADPPRBEVNT	Interrupt status for ADP Probe Event
6	ADPSNSEVNT	Interrupt status for ADP Sense Event
5	ADPTMOUTEVNT	Interrupt status for ADP Timeout Event
4	ADPRSTCMPLTEVNT	Interrupt status for ADP Reset complete Event
3	Reserved	
2	Reserved	
1	Reserved	
0	SER_EVENT	Interrupt status for RAM SER event

3.1.15 IRQ_STATUS_OABS [Offset = 0x0454]

The IRQ_STATUS_OABS Register allows the OTG, ADP, BC (not on K2 devices), and SER interrupt sources to be manually cleared when writing a 1 to a specific bit. A read of this register returns the interrupt event pending value. General actions per bit:

- Write 0: No action
- Write 1: Clear event
- Read 0: No event pending
- Read 1: Event pending

Figure 3-15. IRQ_STATUS_OABS Register

31	28	27	26	25
Reserved		OTGXHCIRUNSTPSETEVNT	OTGDEVRUNSTPSETEVNT	OTGHIBENTRYEVNT
R-0		R/W-0	R/W-0	R/W-0
24		23	22	21
OTGCONIDSTSCHNGEVNT		HRRCONFNOTIFEVNT	HRRINITNOTIFEVNT	OTGADEVIDLEEVNT
R/W-0		R/W-0	R/W-0	R/W-0
20		19	18	17
OTGADEVBHOSTENDEVNT		OTGADEVHOSTEVNT	OTGADEVHNPCHNGEVNT	OTGADEVSRPDETEVNT
R/W-0		R/W-0	R/W-0	R/W-0
16		15	12	11
OTGADEVSESSENDDETEVNT		Reserved		OTGBDEVBHOSTENDEVNT
R/W-0		R-0		R/W-0
10		9	8	7
OTGBDEVHNPCHNGEVNT		OTGBDEVSESSVLDDETEVNT	OTGBDEVVBUSCHNGEVNT	ADPPRBEVNT
R/W-0		R/W-0	R/W-0	R/W-0

6	5	4	3	2	1	0
ADPSNSEVNT	ADPTMOUTEVNT	ADPRSTCMPLTEVNT	Reserved	Reserved	Reserved	SER_EVENT
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-16. IRQ_STATUS_OABS Register Field Descriptions

Bit	Name	Description
31-28	Reserved	
27	OTGXHCIRUNSTPSETEVNT	Interrupt status for OTGXhciRunStpSetEvnt
26	OTGDEVRUNSTPSETEVNT	Interrupt status for OTGDevRunStpSetEvnt
25	OTGHIBENTRYEVNT	Interrupt status for OTGHibEntryEvnt
24	OTGCONIDSTSCHNGEVNT	Interrupt status for OTGConIDStsChngEvnt
23	HRRCONFNOTIFEVNT	Interrupt status for HRRConfNotifEvnt
22	HRRINITNOTIFEVNT	Interrupt status for HRRInitNotifEvnt
21	OTGADEVIDLEEVNT	Interrupt status for OTGADevIdleEvnt
20	OTGADEVBHOSTENDEVNT	Interrupt status for OTGADevHNPChngEvnt
19	OTGADEVHOSTEVNT	Interrupt status for OTGADevHNPChngEvnt
18	OTGADEVHNPCHNGEVNT	Interrupt status for OTGADevHNPChngEvnt
17	OTGADEVSRPDETEVNT	Interrupt status for OTGADevSRPDetEvnt
16	OTGADEVSESSENDDETEVNT	Interrupt status for OTGBDevBHostEndEvnt
15-12	Reserved	
11	OTGBDEVBHOSTENDEVNT	Interrupt status for OTGBDevBHostEndEvnt
10	OTGBDEVHNPCHNGEVNT	Interrupt status for OTGBDevHNPChngEvnt
9	OTGBDEVSESSVLDDETEVNT	Interrupt status for OTGBDevSessVldDetEvnt
8	OTGBDEVVBUSCHNGEVNT	Interrupt status for OTGBDevVBUSChngEvnt

Table 3-16. IRQ_STATUS_OABS Register Field Descriptions (continued)

Bit	Name	Description
7	ADPPRBEVNT	Interrupt status for ADP Probe Event
6	ADPSNSEVNT	Interrupt status for ADP Sense Event
5	ADPTMOUTEVNT	Interrupt status for ADP Timeout Event
4	ADPRSTCMPLTEVNT	Interrupt status for ADP Reset complete Event
3	Reserved	
2	Reserved	
1	Reserved	
0	SER_EVENT	Interrupt status for RAM SER event

3.1.16 IRQ_ENABLE_SET_OABS [Offset = 0x0458]

The IRQ_ENABLE_SET_OABS Register allows the OTG, ADP, BC (not on K2 Devices), and SER interrupt sources to be manually enabled when writing a 1 to a specific bit. A read of this register returns the interrupt event pending value. General actions per bit:

- Write 0: No action
- Write 1: Interrupt enabled
- Read 0: Interrupt disabled
- Read 1: Interrupt enabled

Figure 3-16. IRQ_ENABLE_SET_OABS Register

31	28	27	26	25
Reserved		OTGXHCIRUNSTPSETEVNT	OTGDEVRUNSTPSETEVNT	OTGHIBENTRYEVNT
R-0		R/W-0	R/W-0	R/W-0
24		23	22	21
OTGCONIDSTSCHNGEVNT		HRRCONFNOTIFEVNT	HRRINITNOTIFEVNT	OTGADEVIDLEEVNT
R/W-0		R/W-0	R/W-0	R/W-0
20		19	18	17
OTGADEVBHOSTENDEVNT		OTGADEVHOSTEVNT	OTGADEVHNPCHNGEVNT	OTGADEVSRPDETEVNT
R/W-0		R/W-0	R/W-0	R/W-0
16		15	12	11
OTGADEVSESSENDDETEVNT		Reserved		OTGBDEVBHOSTENDEVNT
R/W-0		R-0		R/W-0
10		9	8	7
OTGBDEVHNPCHNGEVNT		OTGBDEVSESSVLDDETEVNT	OTGBDEVVBUSCHNGEVNT	ADPPRBEVNT
R/W-0		R/W-0	R/W-0	R/W-0

6	5	4	3	2	1	0
ADPSNSEVNT	ADPTMOUTEVNT	ADPRSTCMPLTEVNT	Reserved	Reserved	Reserved	SER_EVENT
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-17. IRQ_ENABLE_SET_OABS Register Field Descriptions

Bit	Name	Description
31-28	Reserved	
27	OTGXHCIRUNSTPSETEVNT	Interrupt enable for OTGXhciRunStpSetEvnt
26	OTGDEVRUNSTPSETEVNT	Interrupt enable for OTGDevRunStpSetEvnt
25	OTGHIBENTRYEVNT	Interrupt enable for OTGHibEntryEvnt
24	OTGCONIDSTSCHNGEVNT	Interrupt enable for OTGConIDStsChngEvnt
23	HRRCONFNOTIFEVNT	Interrupt enable for HRRConfNotifEvnt

Table 3-17. IRQ_ENABLE_SET_OABS Register Field Descriptions (continued)

Bit	Name	Description
22	HRRINITNOTIFEVNT	Interrupt enable for HRRInitNotifEvnt
21	OTGADEVIDLEEVNT	Interrupt enable for OTGADevIdleEvnt
20	OTGADEVBHOSTENDEVNT	Interrupt enable for OTGADevHNPChngEvnt
19	OTGADEVHOSTEVNT	Interrupt enable for OTGADevHNPChngEvnt
18	OTGADEVHNPCHNGEVNT	Interrupt enable for OTGADevHNPChngEvnt
17	OTGADEVSRPDETEVNT	Interrupt enable for OTGADevSRPDetEvnt
16	OTGADEVSESSENDDETEVNT	Interrupt enable for OTGBDevBHostEndEvnt
15-12	Reserved	
11	OTGBDEVBHOSTENDEVNT	Interrupt enable for OTGBDevBHostEndEvnt
10	OTGBDEVHNPCHNGEVNT	Interrupt enable for OTGBDevHNPChngEvnt
9	OTGBDEVSESSVLDDETEVNT	Interrupt enable for OTGBDevSessVldDetEvnt
8	OTGBDEVVBUSCHNGEVNT	Interrupt enable for OTGBDevVBUSChngEvnt
7	ADPPRBEVNT	Interrupt enable for ADP Probe Event
6	ADPSNSEVNT	Interrupt enable for ADP Sense Event
5	ADPTMOUTEVNT	Interrupt enable for ADP Timeout Event
4	ADPRSTCMPLTEVNT	Interrupt enable for ADP Reset complete Event
3	Reserved	
2	Reserved	
1	Reserved	
0	SER_EVENT	Interrupt enable for RAM SER event

3.1.17 IRQ_ENABLE_CLR_OABS [Offset = 0x045C]

The IRQ_ENABLE_CLR_OABS Register allows the OTG, ADP, BC (not on K2 devices), and SER interrupt sources to be manually disabled when writing a 1 to a specific bit. A read of this register returns the interrupt event pending value. General actions per bit:

- Write 0: No action
- Write 1: Disable the event
- Read 0: Interrupt disabled
- Read 1: Interrupt enabled

Figure 3-17. IRQ_ENABLE_CLR_OABS Register

31	28	27	26	25
Reserved		OTGXHCIRUNSTPSETEVNT	OTGDEVRUNSTPSETEVNT	OTGHIBENTRYEVNT
R-0		R/W-0	R/W-0	R/W-0
24		23	22	21
OTGCONIDSTSCHNGEVNT		HRRCONFNOTIFEVNT	HRRINITNOTIFEVNT	OTGADEVIDLEEVNT
R/W-0		R/W-0	R/W-0	R/W-0
20		19	18	17
OTGADEVBHOSTENDEVNT		OTGADEVHOSTEVNT	OTGADEVHNPCHNGEVNT	OTGADEVSRPDETEVNT
R/W-0		R/W-0	R/W-0	R/W-0
16		15	12	11
OTGADEVSESSENDDETEVNT		Reserved	OTGBDEVBHOSTENDEVNT	
R/W-0		R-0	R/W-0	
10		9	8	7
OTGBDEVHNPCHNGEVNT		OTGBDEVSESSVLDDETEVNT	OTGBDEVVBUSCHNGEVNT	ADPPRBEVNT
R/W-0		R/W-0	R/W-0	R/W-0

6	5	4	3	2	1	0
ADPSNSEVNT	ADPTMOUTEVNT	ADPRSTCMPLTEVNT	Reserved	Reserved	Reserved	SER_EVENT
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-18. IRQ_ENABLE_CLR_OABS Register Field Descriptions

Bit	Name	Description
31-28	Reserved	
27	OTGXHCIRUNSTPSETEVNT	Interrupt enable for OTGXhciRunStpSetEvnt
26	OTGDEVRUNSTPSETEVNT	Interrupt enable for OTGDevRunStpSetEvnt
25	OTGHIBENTRYEVNT	Interrupt enable for OTGHibEntryEvnt
24	OTGCONIDSTSCHNGEVNT	Interrupt enable for OTGConIDStsChngEvnt
23	HRRCONFNOTIFEVNT	Interrupt enable for HRRConfNotifEvnt
22	HRRINITNOTIFEVNT	Interrupt enable for HRRInitNotifEvnt
21	OTGADEVIDLEEVNT	Interrupt enable for OTGADevIdleEvnt
20	OTGADEVBHOSTENDEVNT	Interrupt enable for OTGADevHNPChngEvnt
19	OTGADEVHOSTEVNT	Interrupt enable for OTGADevHNPChngEvnt
18	OTGADEVHNPCHNGEVNT	Interrupt enable for OTGADevHNPChngEvnt
17	OTGADEVSRPDETEVNT	Interrupt enable for OTGADevSRPDetEvnt
16	OTGADEVSESSENDDETEVNT	Interrupt enable for OTGBDevBHostEndEvnt
15-12	Reserved	
11	OTGBDEVBHOSTENDEVNT	Interrupt enable for OTGBDevBHostEndEvnt
10	OTGBDEVHNPCHNGEVNT	Interrupt enable for OTGBDevHNPChngEvnt
9	OTGBDEVSESSVLDDETEVNT	Interrupt enable for OTGBDevSessVldDetEvnt
8	OTGBDEVVBUSCHNGEVNT	Interrupt enable for OTGBDevVBUSChngEvnt
7	ADPPRBEVNT	Interrupt enable for ADP Probe Event
6	ADPSNSEVNT	Interrupt enable for ADP Sense Event
5	ADPTMOUTEVNT	Interrupt enable for ADP Timeout Event
4	ADPRSTCMPLTEVNT	Interrupt enable for ADP Reset complete Event
3	Reserved	
2	Reserved	
1	Reserved	
0	SER_EVENT	Interrupt enable for RAM SER event

3.1.18 TXFIFO_DEPTH [Offset = 0x0508]

Actual depth of Tx FIFO RAM (RAM1), in 64-bit words.

Maximum value is GHWPARAMS7.USB3_RAM1_DEPTH. It needs to be initialized at the startup with the actual number of physical RAM locations for the specific configuration.

Figure 3-18. TXFIFO_DEPTH Register

31 16	15 0
Reserved	TXFIFO_DEPTH
R-0	RW-0x1700

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-19. TXFIFO_DEPTH Register Field Descriptions

Bit	Name	Description
31-16	Reserved	
15-0	TXFIFO_DEPTH	The actual Tx FIFO size in 64-bit words

3.1.19 RXFIFO_DEPTH [Offset = 0x050C]

Actual depth of Rx FIFO RAM (RAM2), in 64-bit words.

Maximum value is GHWPARAMS7.USB3_RAM2_DEPTH. It needs to be initialized at the startup with the actual number of physical RAM locations for the specific configuration.

Figure 3-19. RXFIFO_DEPTH Register

31 16	15 0
Reserved	RXFIFO_DEPTH
R-0	RW-0x0400

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-20. RXFIFO_DEPTH Register Field Descriptions

Bit	Name	Description
31-16	Reserved	
15-0	RXFIFO_DEPTH	The actual Rx FIFO size in 64-bit words

3.1.20 SER_CONTROL [Offset = 0x0600]

Soft Error Rate control register sets the enable bits for their respective error correction and/or detection functions. Depending on the application the system may or may not need error detection and/or error correction for each IP. These bits give the option to either enable or disable these features.

Figure 3-20. SER_CONTROL Register

31 3	2 1	0
Reserved	BEC_EN	Reserved
R-0	R/W-0x0002	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-21. SER_CONTROL Register Field Descriptions

Bit	Name	Description
31-3	Reserved	
2-1	BEC_EN	Block error correction and detection enable • 00 = Not enabled • 01 = Enabled, report only errors that have been corrected. • 10 = Enabled, report only errors that have been detected; but not errors that have been corrected • 11 = Enabled, report both errors corrected and detected
0	Reserved	

3.1.21 SER_STATUS [Offset = 0x0604]

Soft Error Rate status register contains many important status fields for the system to diagnose the SER event.

If bec_err is set high, then the block error correction has a non-zero syndrome value. This means that an error has been detected. Single bit errors will be corrected and double bit errors will be detected.

The type of error will be shown in the type_err field. A value of 0 indicates no error has been detected and/or corrected. A value of 1 or 2 indicates either a single or double error has been detected. Values of 1 and 2 will only be valid if a block coder has been implemented (m>3). A value of 3 indicates that the parity detection logic has detected an error event.

The addr_valid field indicates whether the address of the RAM with the error event can be determined. A value of 1 indicates that the address in the SER Address Register is valid. A value of 0 indicates that the address can not be determined. If the address can not be determined, then the system will assume that the entire RAM is corrupted.

The ram_err field indicates which RAM had the error event. This field will be specific to the IP that supports SER and will need to be defined in the respective IP memory map. Each IP can have up to 4096 individual RAMs.

Figure 3-21. SER_STATUS Register

31–28	27–16	15	14
Reserved	RAM_ERR	Reserved	ADDR_VALID
R-0	R/W-0	R-0	R/W-0

13–11	10–8	7–4	3–2	1	0
Reserved	Reserved	Reserved	TYPE_ERR	BEC_ERR	Reserved
R-0	R-0	R-0	R/W-0	R/W-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-22. SER_STATUS Register Field Descriptions

Bit	Name	Description
31-28	Reserved	
27-16	RAM_ERR	Indicates which RAM had the error event. • 0 = No error • 1 = Descriptor RAM 0 EVEN (data bits 31-0) • 2 = Descriptor RAM 0 ODD (data bits 63-32) • 3 = Tx FIFO RAM 1 LO (address 0x0000 to 0x0FFF) • 4 = Tx FIFO RAM 1 HI (address 0x1000 to 0x16FF) • 5 = Rx FIFO RAM 2
15	Reserved	
14	ADDR_VALID	Indicates if the address where the error event occurred can be determined. • 1 = address is known • 0 = address is unknown
13-11	Reserved	

Table 3-22. SER_STATUS Register Field Descriptions (continued)

Bit	Name	Description
10-8	Reserved	
7-4	Reserved	
3-2	TYPE_ERR	Type of error event • 00 = No error event • 01 = Single bit error • 10 = Double bit error • 11 = Reserved
1	BEC_ERR	Block error correction and detection error status • 0 = No error event • 1 = Block error event occurred
0	Reserved	

3.1.22 SER_ADDRESS [Offset = 0x0608]

Soft Error Rate address register indicates the address of the RAM that had the error event

Figure 3-22. SER_ADDRESS Register

31	0
ADDR_ERR	
R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-23. SER_ADDRESS Register Field Descriptions

Bit	Name	Description
31-0	ADDR_ERR	Address of the RAM that had the error event

3.1.23 USBCONFIG [Offset = 0x0614]

USB configuration

Figure 3-23. USBCONFIG Register

31	2	1	0
Reserved		ENDIANMODE	
R-0		R/W-0x0002	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-24. USBCONFIG Register Field Descriptions

Bit	Name	Description
31-2	Reserved	
1-0	ENDIANMODE	Endian mode when the USBSS is in big endian • 2'00 = byte invariant • 2'10 = word invariant

3.1.24 FLADJ [Offset = 0x0704]

Jitter adjustment and other pseudo-static parameters

Figure 3-24. FLADJ Register

31	30	29
CORE_SW_RESET	Reserved	XHCI_REVISION
R/W-0	R-0	R/W-1

28	27	26 21	20 0
HOST_U3_PORT_DISABLE	HOST_U2_PORT_DISABLE	FLADJ_30MHZ	Reserved
R/W-0	R/W-0	R/W-32	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-25. FLADJ Register Field Descriptions

Bit	Name	Description
31	CORE_SW_RESET	Active-high core software reset. Static, i.e. not self-clearing. After clearing, wait for reset completion by polling USBSTS.CNR bit. • NORESET (0) = Reset inactive • RESET (1) = Reset active
30	Reserved	
29	XHCI_REVISION	Switches to the legacy xHCI 0.96 host SW API mode. Changes shall take place under core SW reset (bit 31). • _0_96 (0) = xHCI version 0.96 • _1_0 (1) = xHCI version 1.0 + errata
28	HOST_U3_PORT_DISABLE	USB3 port disable, overriding xHCI driver. • EN (0) = Port can be enabled. • DIS (1) = Port stops reporting connect/disconnect events and remains in disabled state.
27	HOST_U2_PORT_DISABLE	USB2 port disable, overriding xHCI driver. • EN (0) = Port can be enabled. • DIS (1) = Port stops reporting connect/disconnect events and remains in disabled state.
26-21	FLADJ_30MHZ	HS Jitter Adjustment, in 30 MHz periods
20-0	Reserved	

3.1.25 DEBUG_CFG [Offset = 0x0708]

Debug Configuration register controls the selection of USBSS internal signals on the debug interface for observability.

Figure 3-25. DEBUG_CFG Register

31 3	2 0
Reserved	SEL
R-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-26. DEBUG_CFG Register Field Descriptions

Bit	Name	Description
31-3	Reserved	
2-0	SEL	selection of observed local signals • TIELO (0) = Debug output is tied low (32'b0) • UTMI (1) = Debug output is a selection of UTMI (USB2 PHY) interface signals • PIPE (2) = Debug output is a selection of PIPE (USB3 PHY) interface signals • CORE (3) = Debug output is controller core's internal debug signals • TRACE_LO (4) = Debug output is lower 32 bits controller core's internal trace vector, selected by GDBGLSPMUX.TracePortMuxSel • TRACE_HI (5) = Debug output is upper 32 bits controller core's internal trace vector, selected by GDBGLSPMUX.TracePortMuxSel

3.1.26 DEBUG_DATA [Offset = 0x070C]

Data currently visible on DEBUG output (observability) port depends on mode given in DEBUG_CFG.sel , shown only for sel = (1;2;3) all bits are tied low (0) when sel=0.

Figure 3-26. DEBUG_DATA Register

31	30	29	28	27	26	25
DEBUG31	DEBUG30	DEBUG29	DEBUG28	DEBUG27	DEBUG26	DEBUG25
R-0	R-0	R-0	R-0	R-0	R-0	R-0
24	23	22	21	20	19	18
DEBUG24	DEBUG23	DEBUG22	DEBUG21	DEBUG20	DEBUG19	DEBUG18
R-0	R-0	R-0	R-0	R-0	R-0	R-0
17	16	15	14	13	12	11	10
DEBUG17	DEBUG16	DEBUG15	DEBUG14	DEBUG13	DEBUG12	DEBUG11	DEBUG10
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
9	8	7	6	5	4	3	2	1	0
DEBUG9	DEBUG8	DEBUG7	DEBUG6	DEBUG5	DEBUG4	DEBUG3	DEBUG2	DEBUG1	DEBUG0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-27. DEBUG_DATA Register Field Descriptions

Bit	Name	Description
31	DEBUG31	sel=1: utmi_sessend sel=2: pipe_rxstatus[2] sel=3: core_sm2bl_cur_mode
30	DEBUG30	sel=1: utmi_vbusvalid sel=2: pipe_rxstatus[1] sel=3: core_l1_suspend_com_n
29	DEBUG29	sel=1: utmi_bvalid sel=2: pipe_rxstatus[0] sel=3: core_suspend_com_n
28	DEBUG28	sel=1: utmi_avalid sel=2: pipe_elecidle sel=3: core_u2_dssr_state[3]
27	DEBUG27	sel=1: utmi_iddig sel=2: pipe_phystatus sel=3: core_u2_dssr_state[2]
26	DEBUG26	sel=1: utmi_hostdisconnect sel=2: pipe_rxvalid sel=3: core_u2_dssr_state[1]
25	DEBUG25	sel=1: utmi_txbitstuffenableh sel=2: pipe_rxdatak[3] sel=3: core_u2_dssr_state[0]
24	DEBUG24	sel=1: utmi_txbitstuffenable sel=2: pipe_rxdatak[2] sel=3: core_u2mac_txrx_state_1[4]
23	DEBUG23	sel=1: utmi_dischrgvbus sel=2: pipe_rxdatak[1] sel=3: core_u2mac_txrx_state_1[3]
22	DEBUG22	sel=1: utmi_chrgvbus sel=2: pipe_rxdatak[0] sel=3: core_u2mac_txrx_state_1[2]
21	DEBUG21	sel=1: utmi_drvvbus sel=2: pipe_rxpclk sel=3: core_u2mac_txrx_state_1[1]

Table 3-27. DEBUG_DATA Register Field Descriptions (continued)

Bit	Name	Description
20	DEBUG20	sel=1: utmi_dmpulldown sel=2: pipe_rxtermination sel=3: core_u2mac_txrx_state_1[0]
19	DEBUG19	sel=1: utmi_dppulldown sel=2: pipe_txswing sel=3: core_u2mac_txrx_state_0[4]
18	DEBUG18	sel=1: utmi_idpullup sel=2: pipe_txmargin[2] sel=3: core_u2mac_txrx_state_0[3]
17	DEBUG17	sel=1: utmi_linestate[1] sel=2: pipe_txmargin[1] sel=3: core_u2mac_txrx_state_0[2]
16	DEBUG16	sel=1: utmi_linestate[0] sel=2: pipe_txmargin[0] sel=3: core_u2mac_txrx_state_0[1]
15	DEBUG15	sel=1: utmi_opmode[1] sel=2: pipe_txdeemph[1] sel=3: core_u2mac_txrx_state_0[0]
14	DEBUG14	sel=1: utmi_opmode[0] sel=2: pipe_txdeemph[0] sel=3: core_u2_prt_state[4]
13	DEBUG13	sel=1: utmi_termselect sel=2: pipe_powerdown[1] sel=3: core_u2_prt_state[3]
12	DEBUG12	se=1: utmi_xcvrselect[1] sel=2: pipe_powerdown[0] sel=3: core_u2_prt_state[2]
11	DEBUG11	sel=1: utmi_xcvrselect[0] sel=2: pipe_reset_n sel=3: core_u2_prt_state[1]
10	DEBUG10	sel=1: utmi_suspendm sel=2: pipe_rxeqtraining sel=3: core_u2_prt_state[0]
9	DEBUG9	sel=1: utmi_reset sel=2: pipe_rxpolarity sel=3: core_gsts_buserraddvld
8	DEBUG8	sel=1: utmi_rxerror sel=2: pipe_txoneszeros sel=3: debug_mclk_usof_number[0]
7	DEBUG7	sel=1: utmi_rxvalidh sel=2: pipe_txelecidle sel=3: core_ltdb_link_state[3]
6	DEBUG6	sel=1: utmi_rxvalid sel=2: pipe_txdetectrxloopback sel=3: core_ltdb_link_state[2]
5	DEBUG5	sel=1: utmi_rxactive sel=2: pipe_elasticitybuffermode sel=3: core_ltdb_link_state[1]
4	DEBUG4	sel=1: utmi_txready sel=2: pipe_txdatak[3] sel=3: core_ltdb_link_state[0]

Table 3-27. DEBUG_DATA Register Field Descriptions (continued)

Bit	Name	Description
3	DEBUG3	sel=1: utmi_txvalidh sel=2: pipe_txdatak[2] sel=3: core_ltdb_substate[3]
2	DEBUG2	sel=1: utmi_txvalid sel=2: pipe_txdatak[1] sel=3: core_ltdb_substate[2]
1	DEBUG1	sel=1: utmi_databus16_8 sel=2: pipe_txdatak[0] sel=3: core_ltdb_substate[1]
0	DEBUG0	sel=1: utmi_clk sel=2: pipe_txpclk sel=3: core_ltdb_substate[0]

3.1.27 DEV_EBC_EN [Offset = 0x0710]

Enable External Buffer Control (EBC) for selected endpoints. Device mode only. The EBC feature is enabled on devices which implements the Trace over USB feature, such as the K2E and K2L. The EBC feature provides an application hardware-level control to throttle the DMA read/write of data at the packet boundaries without involving software control.

Figure 3-27. DEV_EBC_EN Register

31	30	14	13	0
OUTEP	Reserved	INEP	Reserved	
R/W-0	R-0	R/W-0	R-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-28. DEV_EBC_EN Register Field Descriptions

Bit	Name	Description
31	OUTEP15	Enable EBC HW throttling for OUT EP15, per bit (USB receive). To enable logical OUT EP 15, set DEV_EBC_EN[31] to 1. In DEPCFG command for this EP, DEPCMDPAR1[15] = LimitOutstandingTxDMA should then also be set to 1
30-16	Reserved	
14	INEP15	Enable EBC HW throttling for IN EP15 (USB transmit). To enable logical EP IN14, set DEV_EBC_EN[14] to 1. In DEPCFG command for this EP, DEPCMDPAR1[15] = LimitOutstandingTxDMA should then also be set to 1
13-0	Reserved	

3.1.28 HOST_HUB_CTRL [Offset = 0x0714]

Host Hub Control register is a collection of various input signals that control the xHC controller's Host or Hub interfaces. These signals are used regardless if a Host or Hub is implemented or not.

Figure 3-28. HOST_HUB_CTRL Register

31	12	11	8	7	6	5
Reserved		BUS_FILTER_BYPASS		HUB_PORT_PERM_ATTACH		XHC_BME
R-0x0		R/W-0		R/W-0		R/W-1
4		3		2	1	0
Reserved		HOST_MSI_ENABLE		HOST_PORT_POWER_CONTROL_PRESENT		HUB_PORT_OVERCURRENT
R-0		R/W-1		R/W-1		R/W-0

Legend: R = Read only; W = Write only; - *n* = value after reset; -x, value is indeterminate — see the device-specific data manual

Table 3-29. HOST_HUB_CTRL Register Field Descriptions

Bit	Name	Description
31-12	Reserved	
11-8	BUS_FILTER_BYPASS	The bits mapping: • bit[0]: bypass the filter for vbusvalid • bit[1]: bypass the filter for pipe3_PowerPresent • bit[2]: bypass the filter for avalid, bvalid, and sessend • bit[3]: bypass the filter for iddig The bit setting: • 0 - filter enabled • 1 - filter disabled
7-6	HUB_PORT_PERM_ATTACH	Indicates if the device attached to a downstream port is permanently attached or not. Bit6 is for the USB3 port and the bit7 is for the SuperSpeed port. • 0 - Not permanently attached • 1 - Permanently attached
5	XHC_BME	xHC Bus Master Enable is used to disable the bus mastering capability of the xHC. • 0 - Bus mastering capability is disabled. The host controller cannot use the bus master interface. • 1 - Bus mastering capability is enabled.
4	Reserved	
3	HOST_MSI_ENABLE	In Host mode, it enables the pulse type interrupt signal (one bus clock cycle) on “Interrupt” port instead of level-sensitive interrupt. • 0 - level • 1 - pulse For Device mode, or DRD mode, the interrupt is always level-sensitive
2	HOST_PORT_POWER_CONTROL_PRESENT	This defines the bit[3] of the Capability Parameters (HCCPARAMS) register- change the PPC value through the pin Port Power Control. This indicates whether the host controller implementation includes port power control. • 0 - the port does not have port power switches • 1 - the port has port power switches.
1-0	HUB_PORT_OVERCURRENT	This is the per port Overcurrent indicationof the root-hub ports. Bit0 is the USB2 port and bit1 is for the SuperSpeed port. • 0 - No Overcurrent • 1 - OverCurrent

3.2 USB Core Registers

NOTE: The base address of following registers is device dependent, the specific device’s data sheet should be referred to for it. As an example, on K2K device, it is 0x02680000.

3.2.1 xHCI Capability Registers (host) [Offset = 0]

Table 3-30. xHCI Capability Registers (host)

Offset	Acronym	Register Description	Section
0x10000	CAPLENGTH	Capability Length register	Section 3.2.1.1
0x10004	HCSPARAMS1	Host Controller Structural Parameters 1	Section 3.2.1.2
0x10008	HCSPARAMS2	Host Controller Structural Parameters 2	Section 3.2.1.3
0x1000C	HCSPARAMS3	Host Controller Structural Parameters 3	Section 3.2.1.4
0x10010	HCCPARAMS	Host Controller Capability Parameters	Section 3.2.1.5
0x10014	DBOFF	Doorbell Offset register	Section 3.2.1.6
0x10018	RTSOFF	RunTime Space Offset register	Section 3.2.1.7

3.2.1.1 CAPLENGTH [Offset = 0x10000]

Capability Registers Length + Host Controller Interface Version number

Figure 3-29. CAPLENGTH Register

31	16	15	8	7	0
HCIVERSION		Reserved		CAPLENGTH	
R-0x0100		R-0		R-0x20	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-31. CAPLENGTH Register Field Descriptions

Bit	Name	Description
31-16	HCIVERSION	Host Controller Interface Version (xHCI), in BCD. Set by FLADJ.xHCI_revision field. • _0_96 (150) = xHCI revision 0.96 • _1_00 (256) = xHCI revision 1.0 + errata
15-8	Reserved	
7-0	CAPLENGTH	Capability Register Length: length of the xHCI Capabilities registers bank, in bytes; also the offset of the xHCI Operational registers bank (starting with USBCMD), with respect to xHCI base (i.e. the current CAPLENGTH register)

3.2.1.2 HCSPARAMS1 [Offset = 0x10004]

Host Controller Structural Parameters 1 (xHCI)

Figure 3-30. HCSPARAMS1 Register

31	24	23	19	18	8	7	0
MAXPORTS		Reserved		MAXINTRS		MAXSLOTS	
R-0x02		R-0		R-0x010		R-0x40	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-32. HCSPARAMS1 Register Field Descriptions

Bit	Name	Description
31-24	MAXPORTS	Reg field MAXPORTS
23-19	Reserved	
18-8	MAXINTRS	Reg field MAXINTRS
7-0	MAXSLOTS	Reg field MAXSLOTS

3.2.1.3 HCSPARAMS2 [Offset = 0x10008]

Host Controller Structural Parameters 2 (xHCI)

Figure 3-31. HCSPARAMS2 Register

31	27	26
MAXSCRATCHPADBUFS_LO		SPR
R-0x1		R-0x1

25	21	20	13	12	8	7	4	3	0
MAXSCRATCHPADBUFS_HI		Reserved		Reserved		ERSTMAX		IST	
R-0		R-0		R-0		R--0xF		R-0x1	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-33. HCSPARAMS2 Register Field Descriptions

Bit	Name	Description
31-27	MAXSCRATCHPADBUFS_LO	Max Scratchpad Buffers, lower bits
26	SPR	Scratchpad Restore (SPR). Default = implementation dependent. If Max Scratchpad Buffers is > '0' then this flag indicates whether the xHC uses the Scratchpad Buffers for saving state when executing Save and Restore State operations. If Max Scratchpad Buffers is = '0' then this flag shall be '0'. A value of '1' indicates that the xHC requires the integrity of the Scratchpad Buffer space to be maintained across power events. A value of '0' indicates that the Scratchpad Buffer space may be freed and reallocated between power events.
25-21	MAXSCRATCHPADBUFS_HI	Max Scratchpad Buffers, higher bits
20-13	Reserved	
12-8	Reserved	
7-4	ERSTMAX	Event Ring Segment Table Max
3-0	IST	Isochronous Scheduling Threshold (IST). Default = implementation dependent. The value in this field indicates to system software the minimum distance (in time) that it is required to stay ahead of the host controller while adding TRBs, in order to have the host controller process them at the correct time. The value shall be specified in terms of number of frames/ microframes. If bit [3] of IST is cleared to '0', software can add a TRB no later than IST[2:0] Microframes before that TRB is scheduled to be executed. If bit [3] of IST is set to '1', software can add a TRB no later than IST[2:0] Frames before that TRB is scheduled to be executed.

3.2.1.4 HCSPARAMS3 [Offset = 0x1000C]

Host Controller Structural Parameters 3 (xHCI)

Figure 3-32. HCSPARAMS3 Register

31 16	15 8	7 0
U2_DEVICE_EXIT_LAT	Reserved	U1_DEVICE_EXIT_LAT
R-0x07FF	R-0	R-0x0A

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-34. HCSPARAMS3 Register Field Descriptions

Bit	Name	Description
31-16	U2_DEVICE_EXIT_LAT	U2 Device Exit Latency: Worst case latency to transition from U2 to U0, in us. Applies to all root hub ports. • ZERO (0) = Zero • _1 (1) = Less than 1 us • _2 (2) = Less than 2 us • _2047 (2047) = Less than 2047 us (maximum: greater values are reserved)
15-8	Reserved	
7-0	U1_DEVICE_EXIT_LAT	U1 Device Exit Latency: Worst case latency to transition a root hub Port Link State (PLS) from U1 to U0, in us. • ZERO (0) = Zero • _1 (1) = Less than 1 us • _2 (2) = Less than 2 us • _10 (10) = Less than 10 us (maximum: greater values are reserved)

3.2.1.5 HCCPARAMS [Offset = 0x10010]

Host Controller Capability Parameters (xHCI)

Figure 3-33. HCCPARAMS Register

31	16	15	12	11	10	9	8
XECP		MAXPSASIZE		Reserved		Reserved	PAE
R-0x0298		R-0xF		R-0		R-0	R-0

7	6	5	4	3	2	1	0
NSS	LTC	LHRC	PIND	PPC	CSZ	BNC	AC64
R-0	R-1	R-1	R-0	R-1	R-1	R-0	R-1

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-35. HCCPARAMS Register Field Descriptions

Bit	Name	Description
31-16	XECP	xHCI Extended Capabilties Pointer. 32-bit dword offset, with respect to xHCI base, of the first item of the capability list.
15-12	MAXPSASIZE	Maximum Primary Stream Array Size
11-10	Reserved	
9	Reserved	
8	PAE	Parse All Event data: cf. xHCI 1.0 standard w errata
7	NSS	No Secondary SID Support
6	LTC	Latency Tolerance messaging Capability
5	LHRC	Light HC Reset Capability
4	PIND	Port Indicators
3	PPC	Port Power Control
2	CSZ	Context Size
1	BNC	Bandwidth Negotiation Capability
0	AC64	64-bit Address Capability

3.2.1.6 DBOFF [Offset = 0x10014]

Doorbell Offset (xHCI): Byte offset of doorbell register array (DB[0:N]), with respect to xHCI base (i.e. CAPLENGTH register)

Figure 3-34. DBOFF Register

31	2	1	0
DOORBELL_ARRAY_OFFSET			ZERO
R-0x194			R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-36. DBOFF Register Field Descriptions

Bit	Name	Description
31-2	DOORBELL_ARRAY_OFFSET	Byte address offset MSBits
1-0	ZERO	Byte address offset LSBits, always 0 (offset is 32-bit = 4-byte aligned)

3.2.1.7 RTSOFF [Offset = 0x10018]

RunTime Space Offset (xHCI): Byte offset of runtime register bank (starting with MFINDEX), with respect to xHCI base (i.e. CAPLENGTH register)

Figure 3-35. RTSOFF Register

31	5	4	0
RUNTIME_REG_SPACE_OFFSET			ZERO
R-0x22			R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-37. RTSOFF Register Field Descriptions

Bit	Name	Description
31-5	RUNTIME_REG_SPACE_OFFSET	Byte address offset MSBits
4-0	ZERO	Byte address offset LSBits, always 0 (offset is 32-byte aligned)

3.2.2 xHCI Operational Registers [Offset = 0x10020]

Table 3-38. xHCI Operational Registers (host)

Offset	Acronym	Register Description	Section
0x10020	USBCMD	USB Command register	Section 3.2.2.1
0x10024	USBSTS	USB Status register	Section 3.2.2.2
0x10028	PAGESIZE	Page Size register	Section 3.2.2.3
0x10034	DNCTRL	Device Notification Control register	Section 3.2.2.4
0x10038	CRCR_LO	Command Ring Control register, lower half.	Section 3.2.2.5
0x1003C	CRCR_HI	Command Ring Control register, uppper half.	Section 3.2.2.6
0x10050	DCBAAP_LO	Device Context Base Address Array Pointer register, lower half	Section 3.2.2.7
0x10054	DCBAAP_HI	Device Context Base Address Array Pointer register, upper half	Section 3.2.2.8
0x10058	CONFIG	Configuration register	Section 3.2.2.9

3.2.2.1 USBCMD [Offset = 0x10020]

USB Command Register (xHCI)

Figure 3-36. USBCMD Register

31	12	11	10	9	8
Reserved		EU3S	EWE	CRS	CSS
R-0		R/W-0	R/W-0	R/W-0	R/W-0
7	6 4	3	2	1	0
LHCRST	Reserved	HSEE	INTE	HCRST	R_S
R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-39. USBCMD Register Field Descriptions

Bit	Name	Description
31-12	Reserved	
11	EU3S	Enable U3 MFINDEX Stop
10	EWE	Enable Wrap Event
9	CRS	Controller Restore State: This command is similar to the USBCMD.CRS bit in host mode and initiates the restore process. When software sets this bit to '1', the controller immediately sets DSTS.RSS to '1'. When the controller has finished the restore process, it sets DSTS.RSS to '0'.

Table 3-39. USBCMD Register Field Descriptions (continued)

Bit	Name	Description
8	CSS	Controller Save State: This command is similar to the USBCMD.CSS bit in host mode and initiates the save process. When software sets this bit to '1', the controller immediately sets DSTS.SSS to '1'. When the controller has finished the save process, it sets DSTS.SSS to '0'.
7	LHCRST	Light Host Controller Reset
6-4	Reserved	
3	HSEE	Host System Error Enable
2	INTE	Interrupter Enable
1	HCRST	Host Controller Reset
0	R_S	Run/Stop

3.2.2.2 USBSTS [Offset = 0x10024]

USB Status Register (xHCI)

Figure 3-37. USBSTS Register

31	13	12	11	10	9
Reserved		HCE	CNR	SRE	RSS
R-0		R-0	R-0	R/W-0	R-0

8	7 5	4	3	2	1	0
SSS	Reserved	PCD	EINT	HSE	Reserved	HCH
R-0	R-0	R/W-0	R/W-0	R/W-0	R-0	R-1

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-40. USBSTS Register Field Descriptions

Bit	Name	Description
31-13	Reserved	
12	HCE	Host Controller Error (HCE) - RO. Default = 0. 0' = No internal xHC error conditions exist and '1' = Internal xHC error condition. This flag shall be set to indicate that an internal error condition has been detected which requires software to reset and reinitialize the xHC.
11	CNR	Controller Not Ready (CNR) - RO. Default = '1'. '0' = Ready and '1' = Not Ready. Software shall not write any Doorbell or Operational register of the xHC, other than the USBSTS register, until CNR = '0'. This flag is set by the xHC after a Chip Hardware Reset and cleared when the xHC is ready to begin accepting register writes. This flag shall remain cleared ('0') until the next Chip Hardware Reset.
10	SRE	Save/Restore Error (SRE) -RW1C. Default = '0'. If an error occurs during a Save or Restore operation this bit shall be set to '1'. This bit shall be cleared to '0' when a Save or Restore operation is initiated or when written with '1'. When this register is exposed by a Virtual Function (VF), the VMM determines the state of this bit as a function of the Save/Restore completion status for the selected VF.
9	RSS	Restore State Status: This bit is similar to the USBSTS.RSS in host mode. When the controller has finished the save process, it will complete the command by setting DSTS.RSS to '0'.
8	SSS	Save State Status: This bit is similar to the USBSTS.SSS in host mode. When the controller has finished the save process, it will complete the command by setting DSTS.SSS to '0'.
7-5	Reserved	
4	PCD	Port Change Detect
3	EINT	Event Interrupt
2	HSE	Host System Error
1	Reserved	
0	HCH	Host Controller Halted

3.2.2.3 PAGESIZE [Offset = 0x10028]

Page Size Register (xHCI)

Figure 3-38. PAGESIZE

31	16	15	0
Reserved		PAGE_SIZE	
R-0		R-1	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-41. PAGESIZE Field Descriptions

Bit	Name	Description
31-16	Reserved	
15-0	PAGE_SIZE	Supported system memory page size. When bit #n is set to 1, a page size of 2^(n+12) is supported. • _4KB (1) = 4 kByte page supported

3.2.2.4 DNCTRL [Offset = 0x10034]

Device Notification Control Register (xHCI)

Figure 3-39. DNCTRL Register

31	16	15	14	13	12	11	10	9
Reserved		N15	N14	N13	N12	N11	N10	N9
R/W-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
8		7	6	5	4	3	2	1	0
N8		N7	N6	N5	N4	N3	N2	N1	N0
R/W-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-42. DNCTRL Register Field Descriptions

Bit	Name	Description
31-16	Reserved	
15	N15	Reg field N15
14	N14	Reg field N14
13	N13	Reg field N13
12	N12	Reg field N12
11	N11	Reg field N11
10	N10	Reg field N10
9	N9	Reg field N9
8	N8	Reg field N8
7	N7	Reg field N7
6	N6	Reg field N6
5	N5	Reg field N5
4	N4	Reg field N4
3	N3	Reg field N3
2	N2	Reg field N2
1	N1	Reg field N1
0	N0	Reg field N0

3.2.2.5 CRCR_LO [Offset = 0x10038]

Command Ring Control Register, lower half (xHCI)

Figure 3-40. CRCR_LO Register

31 6	5 4	3	2	1	0
CMD_RING_PNTR	Reserved	CRR	CA	CS	RCS
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-43. CRCR_LO Register Field Descriptions

Bit	Name	Description
31-6	CMD_RING_PNTR	Reg field CMD_RING_PNTR
5-4	Reserved	
3	CRR	Reg field CRR
2	CA	Reg field CA
1	CS	Reg field CS
0	RCS	Reg field RCS

3.2.2.6 CRCR_HI [Offset = 0x1003C]

Command Ring Control Register, upper half (xHCI)

Figure 3-41. CRCR_HI Register

31 0
CMD_RING_PNTR
R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-44. CRCR_HI Register Field Descriptions

Bit	Name	Description
31-0	CMD_RING_PNTR	Reg field COMMAND_RING_POINTER

3.2.2.7 DCBAAP_LO [Offset = 0x10050]

Device Context Base Address Array Pointer, lower half (xHCI)

Figure 3-42. DCBAAP_LO Register

31 6	5 0
DEVICE_CONTEXT_BAAP	Reserved
R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-45. DCBAAP_LO Register Field Descriptions

Bits	Name	Description
31-6	DEVICE_CONTEXT_BAAP	Reg field DEVICE_CONTEXT_BAAP
5-0	Reserved	

3.2.2.8 DCBAAP_HI [Offset = 0x10054]

Device Context Base Address Array Pointer, upper half (xHCI)

Figure 3-43. DCBAAP_HI Register

31	0
DEVICE_CONTEXT_BAAP	
R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-46. DCBAAP_HI Register Field Descriptions

Bit	Name	Description
31-0	DEVICE_CONTEXT_BAAP	Reg field DEVICE_CONTEXT_BAAP

3.2.2.9 CONFIG [Offset = 0x10058]

Configure (xHCI)

Figure 3-44. CONFIG Register

31 8	7 0
Reserved	MAXSLOTSEN
R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-47. CONFIG Register Field Descriptions

Bit	Name	Description
31-8	Reserved	
7-0	MAXSLOTSEN	Reg field MAXSLOTSEN

3.2.3 xHCI Port Registers (host) [Offset = 0x10420]

Table 3-48. xHCI Port Registers (host)

Offset	Acronym	Register Description	Section
0x10420	PORTSC1	Port 1 (USB 2.0) Status and Control register	Section 3.2.3.1
0x10424	PORTPMSC1	Port 1 (USB 2.0) Power Management (LPM) Status and Control register	Section 3.2.3.2
0x10428	PORTLI1	Port 1 (USB 2.0) Link Info register	Section 3.2.3.3
0x10430	PORTSC2	Port 2 (SuperSpeed) Status and Control register	Section 3.2.3.4
0x10434	PORTPMSC2	Port 2 (SuperSpeed) Power Management (LPM) Status and Control register	Section 3.2.3.5
0x10438	PORTLI2	Port 2 (SuperSpeed) Link Info register	Section 3.2.3.6

3.2.3.1 PORTSC1 [Offset = 0x10420]

Port 1 (SuperSpeed) Status and Control (xHCI)

Figure 3-45. PORTSC1 Register

31	30	29 28	27	26	25	24
WPR	DR	Reserved	WOE	WDE	WCE	CAS
R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R-0
23	22	21	20	19	18	17 16
CEC	PLC	PRC	OCC	WRC	PEC	CSC LWS
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0 R/W-0
15 14	13 10	9 8	5	4	3	2 1 0
PIC	PORTSPEED	PP	PLS	PR	OCA	Reserved PED CCS
R/W-0	R-0	R/W-1	R/W-0x5	R/W-0	R-0	R-0 R/W-0 R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-49. PORTSC1 Register Field Descriptions

Bit	Name	Description
31	WPR	Warm Port Reset (WPR) – RW1S/RsvdZ. Default = ‘0’. When software writes a ‘1’ to this bit, the Warm Reset sequence as defined in the USB3 Specification is initiated and the PR flag is set to ‘1’. Once initiated, the PR, PRC, and WRC flags shall reflect the progress of the Warm Reset sequence. This flag shall always return ‘0’ when read. This flag only applies to USB3 protocol ports. For USB2 protocol ports it shall be RsvdZ.
30	DR	Device Removable (DR) - RO. This flag indicates if this port has a removable device attached. ‘1’ = Device is non-removable. ‘0’ = Device is removable.
29-28	Reserved	
27	WOE	Wake on Over-current Enable (WOE) – RWS. Default = ‘0’. Writing this bit to a ‘1’ enables the port to be sensitive to over-current conditions as system wake-up events.
26	WDE	Wake on Disconnect Enable (WDE) – RWS. Default = ‘0’. Writing this bit to a ‘1’ enables the port to be sensitive to device disconnects as system wake-up events.
25	WCE	Wake on Connect Enable (WCE) – RWS. Default = ‘0’. Writing this bit to a ‘1’ enables the port to be sensitive to device connects as system wake-up events.
24	CAS	Cold Attach Status (CAS) – RO. Default = ‘0’. ‘1’ = Far-end Receiver Terminations were detected in the Disconnected state and the Root Hub Port State Machine was unable advance to the Enabled state. Software shall clear this bit by writing a '1' to WPR or the xHC shall clear this bit if CCS transitions to ‘1’. This flag is ‘0’ if PP is ‘0’ or for USB2 protocol ports.
23	CEC	Port Config Error Change (CEC) – RW1CS/RsvdZ. Default = ‘0’. This flag indicates that the port failed to configure its link partner. 0 = No change. 1 = Port Config Error detected. Software shall clear this bit by writing a '1' to it. Note: This flag is valid only for USB3 protocol ports. For USB2 protocol ports this bit shall be RsvdZ.
22	PLC	Port Link State Change (PLC) – RW1CS. Default = ‘0’. This flag is set to ‘1’ due to the following PLS transitions: • U3 -> Resume – Wakeup signaling from a device • Resume -> Recovery -> U0 – Device Resume complete (USB3 protocol ports only) • Resume -> U0 – Device Resume complete (USB2 protocol ports only) • U3 -> Recovery -> U0 – Software Resume complete (USB3 protocol ports only) • U3 -> U0 – Software Resume complete (USB2 protocol ports only) • U2 -> U0 – L1Resume complete (USB2 protocol ports only) • U0 -> U0 – L1 Entry Reject (USB2 protocol ports only) • Any state -> Inactive – Error (USB3 protocol ports only). Note: PLC is asserted only if there is an SS.Inactive.Disconnect.Detect to SS.Inactive.Quiet transition in the LTSSM. Note that this flag shall not be set if the PLS transition was due to software setting PP to ‘0’. Software shall clear this bit by writing a '1' to it.
21	PRC	Port Reset Change (PRC) – RW1CS. Default = ‘0’. This flag is set to ‘1’ due a '1' to '0' transition of Port Reset (PR). e.g. when any reset processing (Warm or Hot) on this port is complete. Note that this flag shall not be set to ‘1’ if the reset processing was forced to terminate due to software clearing PP or PED to '0'. ‘0’ = No change. ‘1’ = Reset complete. Software shall clear this bit by writing a '1' to it.
20	OCC	Over-current Change (OCC) – RW1CS. Default = ‘0’. This bit shall be set to a ‘1’ when there is a ‘0’ to ‘1’ or ‘1’ to ‘0’ transition of Over-current Active (OCA). Software shall clear this bit by writing a ‘1’ to it.
19	WRC	Warm Port Reset Change (WRC) – RW1CS/RsvdZ. Default = ‘0’. This bit is set when Warm Reset processing on this port completes. ‘0’ = No change. ‘1’ = Warm Reset complete. Note that this flag shall not be set to ‘1’ if the Warm Reset processing was forced to terminate due to software clearing PP or PED to '0'. Software shall clear this bit by writing a '1' to it. This bit only applies to USB3 protocol ports. For USB2 protocol ports it shall be RsvdZ.

Table 3-49. PORTSC1 Register Field Descriptions (continued)

Bit	Name	Description
18	PEC	Port Enabled/Disabled Change (PEC) – RW1CS. Default = ‘0’. ‘1’ = change in PED. ‘0’ = No change. Note that this flag shall not be set if the PED transition was due to software setting PP to ‘0’. Software shall clear this bit by writing a ‘1’ to it. For a USB2 protocol port, this bit shall be set to ‘1’ only when the port is disabled due to the appropriate conditions existing at the EOF2 point. For a USB3 protocol port, this bit shall never be set to ‘1’.
17	CSC	Connect Status Change (CSC) – RW1CS. Default = ‘0’. ‘1’ = Change in CCS. ‘0’ = No change. This flag indicates a change has occurred in the port’s Current Connect Status (CCS) or Cold Attach Status (CAS) bits. Note that this flag shall not be set if the CCS transition was due to software setting PP to ‘0’, or the CAS transition was due to software setting WPR to ‘1’. The xHC sets this bit to ‘1’ for all changes to the port device connect status, even if system software has not cleared an existing Connect Status Change. For example, the insertion status changes twice before system software has cleared the changed condition, root hub hardware will be “setting” an already-set bit (i.e., the bit will remain ‘1’). Software shall clear this bit by writing a ‘1’ to it.
16	LWS	Port Link State Write Strobe (LWS) – RW. Default = ‘0’. When this bit is set to ‘1’ on a write reference to this register, this flag enables writes to the PLS field. When ‘0’, write data in PLS field is ignored. Reads to this bit return ‘0’.
15-14	PIC	Port Indicator Control (PIC) – RWS. Default = 0. Writing to these bits has no effect if the Port Indicators (PIND) bit in the HCCPARAMS register is a ‘0’. If PIND bit is a ‘1’, then the bit encodings are: • 0 – Port indicators are off • 1 – Amber • 2 – Green • 3 – Undefined This field is ‘0’ if PP is ‘0’.
13-10	PORTSPEED	Port Speed (Port Speed) – ROS. Default = ‘0’. This field identifies the speed of the attached USB Device. This field is only relevant if a device is attached (CCS = ‘1’) in all other cases this field shall indicate Undefined Speed. 0 – Undefined Speed 1-15 – Protocol Speed ID (PSI) Note: This field is invalid on a USB2 protocol port until after the port is reset.
9	PP	Port Power (PP) – RWS. Default = ‘1’. This flag reflects a port's logical, power control state. Because host controllers can implement different methods of port power switching, this flag may or may not represent whether (VBus) power is actually applied to the port. When PP equals a '0' the port is nonfunctional and shall not report attaches, detaches, or Port Link State (PLS) changes. However, the port shall report over-current conditions when PP = ‘0’ if PPC =‘0’. After modifying PP, software shall read PP and confirm that it is reached its target state before modifying it againh, undefined behavior may occur if this procedure is not followed. 0 = This port is in the Powered-off state. 1 = This port is not in the Powered-off state. If the Port Power Control (PPC) flag in the HCCPARAMS register is '1', then xHC has port power control switches and this bit represents the current setting of the switch ('0' = off, '1' =on). If the Port Power Control (PPC) flag in the HCCPARAMS register is '0', then xHC does not have port power control switches and each port is hard wired to power, and not affected by this bit. When an over-current condition is detected on a powered port, the xHC shall transition the PP bit in each affected port from a ‘1’ to ‘0’ (removing power from the port).

Table 3-49. PORTSC1 Register Field Descriptions (continued)

Bit	Name	Description
8-5	PLS	Port Link State (PLS) – RWS. Default = RxDetect (‘5’). This field is used to power manage the port and reflects its current link state. When the port is in the Enabled state, system software may set the link U state by writing this field. System software may also write this field to force a Disabled to Disconnected state transition of the port. • 0 – The link shall transition to a U0 state from any of the U states. • 2 – USB2 protocol ports only. The link should transition to the U2 State. • 3 – The link shall transition to a U3 state from any of the U states. This action selectively suspends the device connected to this port. While the Port Link State = U3, the hub does not propagate downstream-directed traffic to this port, but the hub shall respond to resume signaling from the port. • 5 – USB3 protocol ports only. If the port is in the Disabled state (PLS = Disabled, PP = 1), then the link shall transition to a RxDetect state and the port shall transition to the Disconnected state, else ignored. • 1,4,6-14 – Ignored. • 15 – USB2 protocol ports only. If the port is in the U3 state (PLS = U3), then the link shall remain in the U3 state and the port shall transition to the U3Exit substate, else ignored. Note: The Port Link State Write Strobe (LWS) shall also be set to ‘1’ to write this field. For USB2 protocol ports: Writing a value of '2' to this field shall request LPM, asserting L1 signaling on the USB2 bus. Software may read this field to determine if the transition to the U2 state was successful. Writing a value of '0' shall deassert L1 signaling on the USB. Writing a value of '1' shall have no effect. The U1 state shall never be reported by a USB2 protocol port. • 0 – Link is in the U0 State • 1 – Link is in the U1 State • 2 – Link is in the U2 State • 3 – Link is in the U3 State (Device Suspended) • 4 – Link is in the Disabled State • 5 – Link is in the RxDetect State • 6 – Link is in the Inactive State • 7 – Link is in the Polling State • 8 – Link is in the Recovery State • 9 – Link is in the Hot Reset State • 10 – Link is in the Compliance Mode State • 11 – Link is in the Test Modef State • 12:14 – Reserved • 15 – Link is in the Resume State This field is undefined if PP = ‘0’. Note: Transitions between different states are not reflected until the transition is complete.
4	PR	Reset. When software writes a ‘1’ to this bit (from a ‘0’) the bus reset sequence is initiated; USB2 protocol ports shall execute the bus reset sequence as defined in the USB2 Spec. USB3 protocol ports shall execute the Hot Reset sequence as defined in the USB3 Spec. PR remains set until reset signaling is completed by the root hub. Note that software shall write a ‘1’ to the this flag to transition a USB2 port from the Polling state to the Enabled state. This flag is ‘0’ if PP is ‘0’.
3	OCA	Over-current Active (OCA) – RO. Default = ‘0’. ‘1’ = This port currently has an over-current condition. ‘0’ = This port does not have an over-current condition. This bit shall automatically transition from a ‘1’ to a ‘0’ when the over-current condition is removed.
2	Reserved	

Table 3-49. PORTSC1 Register Field Descriptions (continued)

Bit	Name	Description
1	PED	Port Enabled/Disabled (PED) – RW1CS. Default = ‘0’. ‘1’ = Enabled. ‘0’ = Disabled. Ports may only be enabled by the xHC. Software cannot enable a port by writing a ‘1’ to this flag. A port may be disabled by software writing a ‘1’ to this flag. This flag shall automatically be cleared to ‘0’ by a disconnect event or other fault condition. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller or bus events. When the port is disabled (PED = ‘0’) downstream propagation of data is blocked on this port, except for reset. For USB2 protocol ports: • When the port is in the Disabled state, software shall reset the port (PR = ‘1’) to transition PED to ‘1’ and the port to the Enabled state. For USB3 protocol ports: • When the port is in the Polling state (after detecting an attach), the port shall automatically transition to the Enabled state and set PED to ‘1’ upon the completion of successful link training. • When the port is in the Disabled state, software shall write a ‘5’ (RxDetect) to the PLS field to transition the port to the Disconnected state. PED shall automatically be cleared to ‘0’ when PR is set to ‘1’, and set to ‘1’ when PR transitions from ‘1’ to ‘0’ after a successful reset. Refer to Port Reset (PR) bit for more information on how the PED bit is managed. Note that when software writes this bit to a ‘1’, it shall also write a ‘0’ to the PR bit. This flag is ‘0’ if PP is ‘0’.
0	CCS	Current Connect Status (CCS) – ROS. Default = ‘0’. ‘1’ = Device is present on port. ‘0’ = No device is present. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change (CSC) bit to be set to ‘1’. This flag is ‘0’ if PP is ‘0’.

3.2.3.2 PORTPMSC1 [Offset = 0x10424]

Port 1 (SuperSpeed) Power Management (LPM) Status and Control (xHCI)

Field structure is protocol-dependent (here: SuperSpeed)

Figure 3-46. PORTPMSC1 Register

31 17	16	15 8	7 0
Reserved	FLA	U2_TIMEOUT	U1_TIMEOUT
R-0	R/W-0	R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-50. PORTPMSC1 Register Field Descriptions

Bit	Name	Description
31-17	Reserved	
16	FLA	Force Link PM Accept (FLA) - RW. Default = ‘0’. When this bit is set to ‘1’, the port shall generate a Set Link Function LMP with the Force_LinkPM_Accept bit asserted. This flag shall be set to ‘0’ by the assertion of PR to ‘1’ or when CCS = transitions from ‘0’ to ‘1’. Writes to this flag have no effect if PP = ‘0’. The Set Link Function LMP is sent by the xHC to the device connected on this port when this bit transitions from ‘0’ to ‘1’. Improper use of the SS Force_LinkPM_Accept functionality can impact the performance of the link significantly. This bit shall only be used for compliance and testing purposes. Software shall ensure that there are no pending packets at the link level before setting this bit. This flag is ‘0’ if PP is ‘0’.

Table 3-50. PORTPMSC1 Register Field Descriptions (continued)

Bit	Name	Description
15-8	U2_TIMEOUT	U2 Timeout – RWS. Default = '0'. Timeout value for U2 inactivity timer. If equal to FFh, the port is disabled from initiating U2 entry. This field shall be set to '0' by the assertion of PR to '1'. The following are permissible values: • 00h – Zero (default) • 01h – 256 μs • 02h – 512 μs • … • FEh – 65.024 ms • FFh – Infinite A U2 Inactivity Timeout LMP shall be sent by the xHC to the device connected on this port when this field is written.
7-0	U1_TIMEOUT	U1 Timeout – RWS. Default = '0'. Timeout value for U1 inactivity timer. If equal to FFh, the port is disabled from initiating U1 entry. This field shall be set to '0' by the assertion of PR to '1'. The following are permissible values: • 00h – Zero (default) • 01h – 1 μs. • 02h – 2 μs. • … • 7Fh – 127 μs. • 80h–FEh – Reserved • FFh – Infinite

3.2.3.3 PORTLI1 [Offset = 0x10428]

Port 1 (SuperSpeed) Link Info (xHCI)

Figure 3-47. PORTLI1 Register

31 16	15 0
Reserved	LINK_ERROR_COUNT
R-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-51. PORTLI1 Register Field Descriptions

Bit	Name	Description
31-16	Reserved	
15-0	LINK_ERROR_COUNT	Link Error Count – RO. Default = '0'. This field returns the number of link errors detected by the port. This value shall be reset to '0' by the assertion of a Chip Hardware Reset, HCRST, when PR transitions from '1' to '0', or when CCS = transitions from '0' to '1'.

3.2.3.4 PORTSC2 [Offset = 0x10430]

Port 2 (USB2.0) Status and Control (xHCI)

Figure 3-48. PORTSC2 Register

31	30	29 28	27	26	25	24
WPR	DR	Reserved	WOE	WDE	WCE	CAS
R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R-0
23	22	21	20	19	18	17	16
CEC	PLC	PRC	OCC	WRC	PEC	CSC	LWS
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15 14	13 10	9 8	5	4	3	2	1	0
PIC	PORTSPEED	PP	PLS	PR	OCA	Reserved	PED	CCS
R/W-0	R-0	R/W-1	R/W-0x4	R/W-0	R-0	R-0	R/W-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-52. PORTSC2 Register Field Descriptions

Bit	Name	Description
31	WPR	Warm Port Reset (WPR) – RW1S/RsvdZ. Default = ‘0’. When software writes a ‘1’ to this bit, the Warm Reset sequence as defined in the USB3 Specification is initiated and the PR flag is set to ‘1’. Once initiated, the PR, PRC, and WRC flags shall reflect the progress of the Warm Reset sequence. This flag shall always return ‘0’ when read. This flag only applies to USB3 protocol ports. For USB2 protocol ports it shall be RsvdZ.
30	DR	Device Removable (DR) - RO. This flag indicates if this port has a removable device attached. ‘1’ = Device is non-removable. ‘0’ = Device is removable.
29-28	Reserved	
27	WOE	Wake on Over-current Enable (WOE) – RWS. Default = ‘0’. Writing this bit to a ‘1’ enables the port to be sensitive to over-current conditions as system wake-up events.
26	WDE	Wake on Disconnect Enable (WDE) – RWS. Default = ‘0’. Writing this bit to a ‘1’ enables the port to be sensitive to device disconnects as system wake-up events.
25	WCE	Wake on Connect Enable (WCE) – RWS. Default = ‘0’. Writing this bit to a ‘1’ enables the port to be sensitive to device connects as system wake-up events.
24	CAS	Cold Attach Status (CAS) – RO. Default = ‘0’. ‘1’ = Far-end Receiver Terminations were detected in the Disconnected state and the Root Hub Port State Machine was unable advance to the Enabled state. Software shall clear this bit by writing a '1' to WPR or the xHC shall clear this bit if CCS transitions to ‘1’. This flag is ‘0’ if PP is ‘0’ or for USB2 protocol ports.
23	CEC	**Port Config Error Change (** CEC) – RW1CS/RsvdZ. Default = ‘0’. This flag indicates that the port failed to configure its link partner. 0 = No change. 1 = Port Config Error detected. Software shall clear this bit by writing a '1' to it. Note: This flag is valid only for USB3 protocol ports. For USB2 protocol ports this bit shall be RsvdZ.
22	PLC	Port Link State Change (PLC) – RW1CS. Default = ‘0’. This flag is set to ‘1’ due to the following PLS transitions: • U3 -> Resume – Wakeup signaling from a device • Resume -> Recovery -> U0 – Device Resume complete (USB3 protocol ports only) • Resume -> U0 – Device Resume complete (USB2 protocol ports only) • U3 -> Recovery -> U0 – Software Resume complete (USB3 protocol ports only) • U3 -> U0 – Software Resume complete (USB2 protocol ports only) • U2 -> U0 – L1Resume complete (USB2 protocol ports only) • U0 -> U0 – L1 Entry Reject (USB2 protocol ports only) • Any state -> Inactive – Error (USB3 protocol ports only). Note: PLC is asserted only if there is an SS.Inactive.Disconnect.Detect to SS.Inactive.Quiet transition in the LTSSM. Note that this flag shall not be set if the PLS transition was due to software setting PP to ‘0’. Software shall clear this bit by writing a '1' to it.
21	PRC	Port Reset Change (PRC) – RW1CS. Default = ‘0’. This flag is set to ‘1’ due a '1' to '0' transition of Port Reset (PR). e.g. when any reset processing (Warm or Hot) on this port is complete. Note that this flag shall not be set to ‘1’ if the reset processing was forced to terminate due to software clearing PP or PED to '0'. ‘0’ = No change. ‘1’ = Reset complete. Software shall clear this bit by writing a '1' to it.
20	OCC	Over-current Change (OCC) – RW1CS. Default = ‘0’. This bit shall be set to a ‘1’ when there is a ‘0’ to ‘1’ or ‘1’ to ‘0’ transition of Over-current Active (OCA). Software shall clear this bit by writing a ‘1’ to it.
19	WRC	Warm Port Reset Change (WRC) – RW1CS/RsvdZ. Default = ‘0’. This bit is set when Warm Reset processing on this port completes. ‘0’ = No change. ‘1’ = Warm Reset complete. Note that this flag shall not be set to ‘1’ if the Warm Reset processing was forced to terminate due to software clearing PP or PED to '0'. Software shall clear this bit by writing a '1' to it. This bit only applies to USB3 protocol ports. For USB2 protocol ports it shall be RsvdZ.
18	PEC	Port Enabled/Disabled Change (PEC) – RW1CS. Default = ‘0’. ‘1’ = change in PED. ‘0’ = No change. Note that this flag shall not be set if the PED transition was due to software setting PP to ‘0’. Software shall clear this bit by writing a ‘1’ to it. For a USB2 protocol port, this bit shall be set to ‘1’ only when the port is disabled due to the appropriate conditions existing at the EOF2 point. For a USB3 protocol port, this bit shall never be set to ‘1’.
17	CSC	Connect Status Change (CSC) – RW1CS. Default = ‘0’. ‘1’ = Change in CCS. ‘0’ = No change. This flag indicates a change has occurred in the port’s Current Connect Status (CCS) or Cold Attach Status (CAS) bits. Note that this flag shall not be set if the CCS transition was due to software setting PP to ‘0’, or the CAS transition was due to software setting WPR to ‘1’. The xHC sets this bit to ‘1’ for all changes to the port device connect status, even if system software has not cleared an existing Connect Status Change. For example, the insertion status changes twice before system software has cleared the changed condition, root hub hardware will be “setting” an already-set bit (i.e., the bit will remain ‘1’). Software shall clear this bit by writing a ‘1’ to it.
16	LWS	Port Link State Write Strobe (LWS) – RW. Default = ‘0’. When this bit is set to ‘1’ on a write reference to this register, this flag enables writes to the PLS field. When ‘0’, write data in PLS field is ignored. Reads to this bit return ‘0’.

Table 3-52. PORTSC2 Register Field Descriptions (continued)

Bit	Name	Description
15-14	PIC	Port Indicator Control (PIC) – RWS. Default = 0. Writing to these bits has no effect if the Port Indicators (PIND) bit in the HCCPARAMS register is a ‘0’. If PIND bit is a ‘1’, then the bit encodings are: • 0 – Port indicators are off • 1 – Amber • 2 – Green • 3 – Undefined This field is ‘0’ if PP is ‘0’.
13-10	PORTSPEED	Port Speed (Port Speed) – ROS. Default = ‘0’. This field identifies the speed of the attached USB Device. This field is only relevant if a device is attached (CCS = ‘1’) in all other cases this field shall indicate Undefined Speed. 0 – Undefined Speed 1-15 – Protocol Speed ID (PSI) Note: This field is invalid on a USB2 protocol port until after the port is reset.
9	PP	Port Power (PP) – RWS. Default = ‘1’. This flag reflects a port's logical, power control state. Because host controllers can implement different methods of port power switching, this flag may or may not represent whether (VBus) power is actually applied to the port. When PP equals a '0' the port is nonfunctional and shall not report attaches, detaches, or Port Link State (PLS) changes. However, the port shall report over-current conditions when PP = ‘0’ if PPC =‘0’. After modifying PP, software shall read PP and confirm that it is reached its target state before modifying it againh, undefined behavior may occur if this procedure is not followed. 0 = This port is in the Powered-off state. 1 = This port is not in the Powered-off state. If the Port Power Control (PPC) flag in the HCCPARAMS register is '1', then xHC has port power control switches and this bit represents the current setting of the switch ('0' = off, '1' =on). If the Port Power Control (PPC) flag in the HCCPARAMS register is '0', then xHC does not have port power control switches and each port is hard wired to power, and not affected by this bit. When an over-current condition is detected on a powered port, the xHC shall transition the PP bit in each affected port from a ‘1’ to ‘0’ (removing power from the port).

Table 3-52. PORTSC2 Register Field Descriptions (continued)

Bit	Name	Description
8-5	PLS	Port Link State (PLS) – RWS. Default = RxDetect (‘5’). This field is used to power manage the port and reflects its current link state. When the port is in the Enabled state, system software may set the link U state by writing this field. System software may also write this field to force a Disabled to Disconnected state transition of the port. • 0 – The link shall transition to a U0 state from any of the U states. • 2 – USB2 protocol ports only. The link should transition to the U2 State. • 3 – The link shall transition to a U3 state from any of the U states. This action selectively suspends the device connected to this port. While the Port Link State = U3, the hub does not propagate downstream-directed traffic to this port, but the hub shall respond to resume signaling from the port. • 5 – USB3 protocol ports only. If the port is in the Disabled state (PLS = Disabled, PP = 1), then the link shall transition to a RxDetect state and the port shall transition to the Disconnected state, else ignored. • 1,4,6-14 – Ignored. • 15 – USB2 protocol ports only. If the port is in the U3 state (PLS = U3), then the link shall remain in the U3 state and the port shall transition to the U3Exit substate, else ignored. Note: The Port Link State Write Strobe (LWS) shall also be set to ‘1’ to write this field. For USB2 protocol ports: Writing a value of '2' to this field shall request LPM, asserting L1 signaling on the USB2 bus. Software may read this field to determine if the transition to the U2 state was successful. Writing a value of '0' shall deassert L1 signaling on the USB. Writing a value of '1' shall have no effect. The U1 state shall never be reported by a USB2 protocol port. • 0 – Link is in the U0 State • 1 – Link is in the U1 State • 2 – Link is in the U2 State • 3 – Link is in the U3 State (Device Suspended) • 4 – Link is in the Disabled State • 5 – Link is in the RxDetect State • 6 – Link is in the Inactive State • 7 – Link is in the Polling State • 8 – Link is in the Recovery State • 9 – Link is in the Hot Reset State • 10 – Link is in the Compliance Mode State • 11 – Link is in the Test Modef State • 12:14 – Reserved • 15 – Link is in the Resume State This field is undefined if PP = ‘0’. Note: Transitions between different states are not reflected until the transition is complete.
4	PR	Reset. When software writes a ‘1’ to this bit (from a ‘0’) the bus reset sequence is initiated; USB2 protocol ports shall execute the bus reset sequence as defined in the USB2 Spec. USB3 protocol ports shall execute the Hot Reset sequence as defined in the USB3 Spec. PR remains set until reset signaling is completed by the root hub. Note that software shall write a ‘1’ to the this flag to transition a USB2 port from the Polling state to the Enabled state. This flag is ‘0’ if PP is ‘0’.
3	OCA	Over-current Active (OCA) – RO. Default = ‘0’. ‘1’ = This port currently has an over-current condition. ‘0’ = This port does not have an over-current condition. This bit shall automatically transition from a ‘1’ to a ‘0’ when the over-current condition is removed.
2	Reserved	

Table 3-52. PORTSC2 Register Field Descriptions (continued)

Bit	Name	Description
1	PED	Port Enabled/Disabled (PED) – RW1CS. Default = ‘0’. ‘1’ = Enabled. ‘0’ = Disabled. Ports may only be enabled by the xHC. Software cannot enable a port by writing a ‘1’ to this flag. A port may be disabled by software writing a ‘1’ to this flag. This flag shall automatically be cleared to ‘0’ by a disconnect event or other fault condition. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller or bus events. When the port is disabled (PED = ‘0’) downstream propagation of data is blocked on this port, except for reset. For USB2 protocol ports: • When the port is in the Disabled state, software shall reset the port (PR = ‘1’) to transition PED to ‘1’ and the port to the Enabled state. For USB3 protocol ports: • When the port is in the Polling state (after detecting an attach), the port shall automatically transition to the Enabled state and set PED to ‘1’ upon the completion of successful link training. • When the port is in the Disabled state, software shall write a ‘5’ (RxDetect) to the PLS field to transition the port to the Disconnected state. PED shall automatically be cleared to ‘0’ when PR is set to ‘1’, and set to ‘1’ when PR transitions from ‘1’ to ‘0’ after a successful reset. Refer to Port Reset (PR) bit for more information on how the PED bit is managed. Note that when software writes this bit to a ‘1’, it shall also write a ‘0’ to the PR bit. This flag is ‘0’ if PP is ‘0’.
0	CCS	Current Connect Status (CCS) – ROS. Default = ‘0’. ‘1’ = Device is present on port. ‘0’ = No device is present. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change (CSC) bit to be set to ‘1’. This flag is ‘0’ if PP is ‘0’.

3.2.3.5 PORTPMSC2 [Offset = 0x10434]

Port 2 (USB2.0) Power Management Status and Control (xHCI)

Field structure is protocol-dependent (here: USB2.0)

Figure 3-49. PORTPMSC2 Register

31 28	27 17	16	15 8	7 4	3	2 0
PORT_TEST_CONTROL	Reserved	HLE	L1_DEVICE_SLOT	HIRD	RWE	L1S
R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-53. PORTPMSC2 Register Field Descriptions

Bit	Name	Description
31-28	PORT_TEST_CONTROL	Port Test Control – RW. Default = ‘0’. When this field is ‘0’, the port is NOT operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value. A non-zero Port Test Control value is only valid to a port that is in the Powered-Off state (PLS =Disabled). If the port is not in this state, the xHC shall respond with the Port Test Control field set to Port Test Control Error. The encoding of the Test Mode bits for a USB2 protocol port are: • 0 – Test mode not enabled • 1 – Test J_STATE • 2 – Test K_STATE • 3 – Test SE0_NAK • 4 – Test Packet • 5 – Test FORCE_ENABLE • 6-14 – Reserved. • 15 – Port Test Control Error.
27-17	Reserved	

Table 3-53. PORTPMSC2 Register Field Descriptions (continued)

Bit	Name	Description
16	HLE	Hardware LPM Enable (HLE) - RW. Default = '0'. If this bit is set to '1', then hardware controlled LPM shall be enabled for this port. If the USB2 Hardware LMP Capability is not supported (HLC = '0') this field shall be RsvdZ.
15-8	L1_DEVICE_SLOT	L1 Device Slot - RW. Default = '0'. System software sets this field to indicate the ID of the Device Slot associated with the device directly attached to the Root Hub port. A value of '0' indicates no device is present. The xHC uses this field to lookup information necessary to generate the LMP Token packet.
7-4	HIRD	Host Initiated Resume Duration (HIRD) - RW. Default = '0'. System software sets this field to indicate to the recipient device how long the xHC will drive resume if it (the xHC) initiates an exit from L1. The HIRD value is encoded as follows: The value of 0000b is interpreted as 50 μs. Each incrementing value up adds 75 μs to the previous value. For example, 0001b is 125 μs, 0010b is 200 μs and so on. Based on this rule, the maximum value resume drive time is at encoding value 1111b which represents 1.2ms.
3	RWE	Remote Wake Enable (RWE) - RW. Default = '0'. System software sets this flag to enable or disable the device for remote wake from L1. The value of this flag shall temporarily (while in L1) override the current setting of the Remote Wake feature set by the standard Set/ClearFeature() commands defined in Universal Serial Bus Specification, revision 2.0, Chapter 9.
2-0	L1S	L1 Status (L1S) - RO. Default = 0. This field is used by software to determine whether an L1-based suspend request (LMP transaction) was successful, specifically: • 0 – Invalid - This field shall be ignored by software. • 1 – Success - Port successfully transitioned to L1 (ACK) • 2 – Not Yet - Device is unable to enter L1 at this time (NYET) • 3 – Not Supported - Device does not support L1 transitions (STALL) • 4 – Timeout/Error - Device failed to respond to the LPM Transaction or an error occurred • 5-7 – Reserved The value of this field is only valid when the port resides in the L0 or L1 state (PLS = '0' or '2').

3.2.3.6 PORTLI2 [Offset = 0x10438]

Port 2 (USB2.0) Link Info (xHCI)

Figure 3-50. PORTLI2 Register

31 16	15 0
Reserved	LINK_ERROR_COUNT
R-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-54. PORTLI2 Register Field Descriptions

Bit	Name	Description
31-16	Reserved	
15-0	LINK_ERROR_COUNT	Link Error Count – RO. Default = '0'. This field returns the number of link errors detected by the port. This value shall be reset to '0' by the assertion of a Chip Hardware Reset, HCRST, when PR transitions from '1' to '0', or when CCS = transitions from '0' to '1'.

3.2.4 xHCI Runtime Registers (host) [Offset = 0x10440]

Table 3-55. xHCI Runtime Registers (host)

Offset	Acronym	Register Description	Section
0x10440	MFINDEX	Microframe Index register	Section 3.2.4.1

3.2.4.1 MFINDEX [Offset = 0x10440]

Microframe Index (xHCI)

Figure 3-51. MFINDEX Register

31 14	13 0
Reserved	MICROFRAME_INDEX
R-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-56. MFINDEX Register Field Descriptions

Bit	Name	Description
31-14	Reserved	
13-0	MICROFRAME_INDEX	Reg field MICROFRAME_INDEX

3.2.5 xHCI Interrupter Registers [Offset = 0x10460]

Table 3-57. xHCI Interrupter Registers

Offset	Acronym	Register Description	Section
0x10460+0x20*N	IMAN[N]	Interrupter Management registers	Section 3.2.5.1
0x10464+0x20*N	IMOD[N]	Interrupter Moderation registers	Section 3.2.5.2
0x10468+0x20*N	ERSTSZ[N]	Event Ring Segment Table Size registers	Section 3.2.5.3
0x10470+0x20*N	ERSTBA_LO[N]	Event Ring Segment Table Base Address, lower half, registers	Section 3.2.5.4
0x10474+0x20*N	ERSTBA_HI[N]	Event Ring Segment Table Base Address, upper half, registers	Section 3.2.5.5
0x10478+0x20*N	ERDP_LO[N]	Event Ring Dequeue Pointer, lower half, registers	Section 3.2.5.6
0x1047C+0x20*N	ERDP_HI[N]	Event Ring Dequeue Pointer, upper half, registers	Section 3.2.5.7

3.2.5.1 IMAN[N] [Offset = 0x10460]

Interrupter Management (xHCI)

Figure 3-52. IMAN Register

31 2	1	0
Reserved	IE	IP
R-0	R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-58. IMAN Register Field Descriptions

Bit	Name	Description
31-2	Reserved	
1	IE	Interrupt Enable. • DIS (0) = Interrupt disabled • EN (1) = Interrupt enabled
0	IP	Interrupt Pending. Set (to 1) when: IE=1, IMODC=0, associated Event Ring is not empty, EHB=0. • PENDING (1) = IRQ pending • IDLE (0) = No IRQ pending

3.2.5.2 IMOD[N] [Offset = 0x10464+0x20*N, N=0-15]

Interrupter Moderation (xHCI)

Figure 3-53. IMOD Register

31	16	15	0
IMODC		IMODI	
R/W-0		R/W-4000	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-59. IMOD Register Field Descriptions

Bit	Name	Description
31-16	IMODC	Interrupt Moderation Counter: Loaded to IMODI whenever IP is cleared to 0, counts down to 0, and stops. IRQ is generated when counter is 0, Event Ring is not empty, IE=1, IP=1, EHB=0. May be directly written at any time to alter the interrupt rate.
15-0	IMODI	Interrupt Moderation Interval: Minimum inter-IRQ interval, in 250ns increments. • DIS (0) = IRQ throttling disabled: IRQs generated immediately. • _250NS (1) = 250 ns minimum interval • _1MS1 (4000) = 1 ms interval (default)

3.2.5.3 ERSTSZ[N] [Offset = 0x10468+0x20*N, N=0-15]

Event Ring Segment Table Size (xHCI)

Figure 3-54. ERSTSZ Register

31	16	15	0
Reserved		ERS_TABLE_SIZE	
R/W-0		R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-60. ERSTSZ Register Field Descriptions

Bit	Name	Description
31-16	Reserved	
15-0	ERS_TABLE_SIZE	Event Ring Segment Table Size – RW. Default = '0'. This field identifies the number of valid Event Ring Segment Table entries in the Event Ring Segment Table pointed to by the Event Ring Segment Table Base Address register. The maximum value supported by an xHC implementation for this register is defined by the ERST Max field in the HCSPARAMS2 register. For Secondary Interrupters: Writing a value of '0' to this field disables the Event Ring. Any events targeted at this Event Ring when it is disabled shall result in undefined behavior of the Event Ring. For the Primary Interrupter: Writing a value of '0' to this field shall result in undefined behavior of the Event Ring. The Primary Event Ring cannot be disabled.

3.2.5.4 ERSTBA_LO[N] [Offset = 0x10470+0x20*N, N=0-15]

Event Ring Segment Table Base Address, lower half (xHCI)

Figure 3-55. ERSTBA_LO Register

31	6	5	0
ERS_TABLE_BAR		Reserved	
R/W-0		R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-61. ERSTBA_LO Register Field Descriptions

Bit	Name	Description
31-6	ERS_TABLE_BAR	Event Ring Segment Table Base Address Register – RW. Default = ‘0’. This field defines the high order bits of the start address of the Event Ring Segment Table. Writing this register sets the Event Ring State Machine:EREP Advancement to the Start state. This field shall not be modified if HCHalted (HCH) = ‘0’.
5-0	Reserved	

3.2.5.5 ERSTBA_HI[N] [Offset = 0x10474+0x20*N, N=0-15]

Event Ring Segment Table Base Address, upper half (xHCI)

Figure 3-56. ERSTBA_HI Register

31	0
ERS_TABLE_BAR	
R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-62. ERSTBA_HI Register Field Descriptions

Bit	Name	Description
31-0	ERS_TABLE_BAR	Event Ring Segment Table Base Address Register – RW. Default = ‘0’. This field defines the high order bits of the start address of the Event Ring Segment Table. Writing this register sets the Event Ring State Machine:EREP Advancement to the Start state. This field shall not be modified if HCHalted (HCH) = ‘0’.

3.2.5.6 ERDP_LO[N] [Offset = 0x10478+0x20*N, N=0-15]

Event Ring Dequeue Pointer, lower half (xHCI)

Figure 3-57. ERDP_LO Register

31	4	3	2	0
ERD_PNTR		EHB	DESI	
R/W-0		R/W-0	R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-63. ERDP_LO Register Field Descriptions

Bit	Name	Description
31-4	ERD_PNTR	Event Ring Dequeue Pointer - RW. Default = ‘0’. This field defines the high order bits of the 64- bit address of the current Event Ring Dequeue Pointer.
3	EHB	Event Handler Busy (EHB) - RW1C. Default = ‘0’. This flag shall be set to ‘1’ when the IP bit is set to ‘1’ and cleared to ‘0’ by software when the Dequeue Pointer register is written.
2-0	DESI	Dequeue ERST Segment Index (DESI). Default = ‘0’. This field may be used by the xHC to accelerate checking the Event Ring full condition. This field is written with the low order 3 bits of the offset of the ERST entry which defines the Event Ring segment that the Event Ring Dequeue Pointer resides in.

3.2.5.7 ERDP_HI[N] [Offset = 0x1047C+0x20*N, N=0-15]

Event Ring Dequeue Pointer, upper half (xHCI)

Figure 3-58. ERDP_HI Register

31	0
ERD_PNTR	
R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-64. ERDP_HI Register Field Descriptions

Bit	Name	Description
31-0	ERD_PNTR	Event Ring Dequeue Pointer - RW. Default = '0'. This field defines the high order bits of the 64- bit address of the current Event Ring Dequeue Pointer.

3.2.6 xHCI Doorbell Registers [Offset = 0x10660]

Table 3-65. xHCI Doorbell Registers

Offset	Acronym	Register Description	Section
0x10480 + 4*N	DB[N]	DB	Section 3.2.6.1

3.2.6.1 DB[N] [Offset = 0x10660+4*N, N=0-63]

Doorbell (xHCI)

Figure 3-59. DB Register

31 16	15 8	7 0
DB_STREAM_ID	Reserved	DB_TARGET
R/W-0	R-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-66. DB Register Field Descriptions

Bit	Name	Description
31-16	DB_STREAM_ID	DB Stream ID - RW. Doorbell Stream ID. If the endpoint of a Device Context Doorbell defines Streams, then this field shall be used to identify which Stream of the endpoint the doorbell reference is targeting. System software is responsible for ensuring that the value written to this field is valid. If the endpoint defines Streams (MaxPStreams > 0), then 0, 65535 (No Stream) and 65534 (Prime) are reserved Stream ID values and shall not be written to this field. If the endpoint does not define Streams (MaxPStreams = 0) and a non-'0' value is written to this field, the doorbell reference shall be ignored. This field only applies to Device Context Doorbells and shall be cleared to '0' for Host Controller Command Doorbells. This field returns '0' when read.
15-8	Reserved	

Table 3-66. DB Register Field Descriptions (continued)

Bit	Name	Description
7-0	DB_TARGET	DB Target – RW. Doorbell Target. This field defines the target of the doorbell reference. The table below defines the xHC notification that is generated by ringing the doorbell. Note that Doorbell Register 0 is dedicated to Command Ring and decodes this field differently than the other Doorbell Registers. Device Context Doorbells (1-255) • 0 – Reserved • 1 – Control EP 0 Enqueue Pointer Update • 2 – EP 1 OUT Enqueue Pointer Update • 3 – EP 1 IN Enqueue Pointer Update • 4 – EP 2 OUT Enqueue Pointer Update • 5 – EP 2 IN Enqueue Pointer Update • … ... • 30 – EP 15 OUT Enqueue Pointer Update • 31 – EP 15 IN Enqueue Pointer Update • 32:247 – Reserved • 248:255 – Vendor Defined Host Controller Doorbell (0) • 0 – Command Doorbell • 1:247 – Reserved • 248:255 – Vendor Defined This field returns '0' when read and should be treated as "undefined" by software. When the Command Doorbell is written, the DB Stream ID field shall be cleared to '0'.

3.2.7 xHCI Extended Capabilities Registers [Offset = 0x10A60]

Table 3-67. xHCI Extended Capabilities Registers

Offset	Acronym	Register Description	Section
0x10A60	USBLEGSUP	USB Legacy Support Capability register	Section 3.2.7.1
0x10A64	USBLEGCTLSTS	USB Legacy Control and Status register	Section 3.2.7.2
0x10A70	SUPTPRT2_DW0	Supported Protocol capability register for USB2, 32-bit dword #0	Section 3.2.7.3
0x10A74	SUPTPRT2_DW1	Supported Protocol capability register for USB2, 32-bit dword #1	Section 3.2.7.4
0x10A78	SUPTPRT2_DW2	Supported Protocol capability register for USB2, 32-bit dword #2	Section 3.2.7.5
0x10A80	SUPTPRT3_DW0	Supported Protocol capability register for USB3, 32-bit dword #0	Section 3.2.7.6
0x10A84	SUPTPRT3_DW1	Supported Protocol capability register for USB3, 32-bit dword #1	Section 3.2.7.7
0x10A88	SUPTPRT3_DW2	Supported Protocol capability register for USB3, 32-bit dword #2	Section 3.2.7.8

3.2.7.1 USBLEGSUP [Offset = 0x10A60]

USB Legacy Support Capability

Figure 3-60. USBLEGSUP Register

31 25	24	23 17	16	15 8	7 0
Reserved	HCOOS	Reserved	HCBOS	NCP	ECID
R-0	R/W-0	R-0	R/W-0	R-4	R-1

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-68. USBLEGSUP Register Field Descriptions

Bit	Name	Description
31-25	Reserved	
24	HCOOS	HC OS Owned Semaphore
23-17	Reserved	
16	HCBOS	HC BIOS Owned Semaphore
15-8	NCP	Next Capability Pointer: 32-bit dword offset of the next capability. • EOL (0) = End of capability list
7-0	ECID	Extended Capability ID code (descriptor size, in bytes) • LEGACY (1) = USB legacy support (8) • PROT (2) = Supported protocol: MajRev.MinRev (12) • DBC (10) = USB debug capability (56)

3.2.7.2 USBLEGCTLSTS [Offset = 0x10A64]

USB Legacy Control / Status

Figure 3-61. USBLEGCTLSTS Register

31	30	29	28 21	20	19 17
SB	SPC	SOOC	Reserved	SHSE	Reserved
R/W-0	R/W-0	R/W-0	R-0	R-0	R-0

16	15	14	13	12 5	4	3 1	0
SEI	SBE	SPCE	SOOE	Reserved	SHSEE	Reserved	USE
R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-69. USBLEGCTLSTS Register Field Descriptions

Bit	Name	Description
31	SMI_ON_BAR	Reg field SMI_ON_BAR
30	SMI_ON_PCI	Reg field SMI_ON_PCI COMMAND
29	SMI_ON_OS	Reg field SMI_ON_OS OWNERSHIP CHANGE
28-21	Reserved	
20	SMI_ON_HOST	Reg field SMI_ON_HOST SYSTEM ERROR
19-17	Reserved	
16	SMI_ON_EVENT	Reg field SMI_ON_EVENT INTERRUPT
15	SMI_ON_BAR_E	Reg field SMI_ON_BAR ENABLE
14	SMI_ON_PCI_E	Reg field SMI_ON_PCI COMMAND ENABLE
13	SMI_ON_OS_E	Reg field SMI_ON_OS OWNERSHIP ENABLE
12-5	Reserved	
4	SMI_ON_HOST_E	Reg field SMI_ON_HOST SYSTEM ERROR ENABLE
3-1	Reserved	
0	USB_SMI_ENABLE	Reg field USB_SMI_ENABLE

3.2.7.3 SUPTPRT2_DW0 [Offset = 0x10A70]

Supported protocol capability USB2, 32-bit dword #0

Figure 3-62. SUPTPRT2_DW0 Register

31 24	23 16	15 8	7 0
MAJREV	MINREV	NCP	ECID
R-2	R-0	R-4	R-2

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-70. SUPTPRT2_DW0 Register Field Descriptions

Bit	Name	Description
31-24	MAJREV	Major Revision, BCD-encoded
23-16	MINREV	Minor Revision, BCD-encoded
15-8	NCP	Next Capability Pointer: 32-bit dword offset of the next capability. • EOL (0) = End of capability list
7-0	ECID	Extended Capability ID code (descriptor size, in bytes) • LEGACY (1) = USB legacy support (8) • PROT (2) = Supported protocol: MajRev.MinRev (12) • DBC (10) = USB debug capability (56)

3.2.7.4 SUPTPRT2_DW1 [Offset = 0x10A74]

Supported protocol capability USB2, 32-bit dword #1: Name String "USB "

Figure 3-63. SUPTPRT2_DW1 Register

31 24	23 16	15 8	7 0
CHAR3	CHAR2	CHAR1	CHAR0
R- 0x20	R-0x42	R-0x53	R-0x55

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-71. SUPTPRT2_DW1 Register Field Descriptions

Bit	Name	Description
31-24	CHAR3	ASCII " " (space)
23-16	CHAR2	ASCII "B"
15-8	CHAR1	ASCII "S"
7-0	CHAR0	ASCII "U"

3.2.7.5 SUPTPRT2_DW2 [Offset = 0x10A78]

Supported protocol capability USB2, 32-bit dword #2

Figure 3-64. SUPTPRT2_DW2 Register

31 28	27 20	19	18	17	16	15 8	7 0
PSIC	Reserved	HLC	IHI	HSO	Reserved	CPC	CPO
R-0	R-0	R-1	R-0	R-0	R-0	R-1	R-1

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-72. SUPTPRT2_DW2 Register Field Descriptions

Bit	Name	Description
31-28	PSIC	Port Speed ID Count. Reserved in xHCI 0.96 • ALL (0) = USB2 High/Full/Low-speeds supported
27-20	Reserved	
19	HLC	Hardware LPM Capability.
18	IHI	Integrated Hub Implemented.
17	HSO	High-Speed Only
16	Reserved	
15-8	CPC	Compatible Port Count: Number of consecutive ports of the root hub that support this protocol, from CPO to CPO+CPC-1
7-0	CPO	Compatible Port Offset: Starting port number of root hub port(s) that support this protocol.

3.2.7.6 SUPTPRT3_DW0 [Offset = 0x10A80]

Supported protocol capability USB3, 32-bit dword #0

Figure 3-65. SUPTPRT3_DW0 Register

31	24	23	16	15	8	7	0
MAJREV		MINREV		NCP		ECID	
R-3		R-0		R-0		R-2	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-73. SUPTPRT3_DW0 Register Field Descriptions

Bit	Name	Description
31-24	MAJREV	Major Revision, BCD-encoded
23-16	MINREV	Minor Revision, BCD-encoded
15-8	NCP	Next Capability Pointer: 32-bit dword offset of the next capability. • EOL (0) = End of capability list
7-0	ECID	Extended Capability ID code (descriptor size, in bytes) • LEGACY (1) = USB legacy support (8) • PROT (2) = Supported protocol: MajRev.MinRev (12) • DBC (10) = USB debug capability (56)

3.2.7.7 SUPTPRT3_DW1 [Offset = 0x10A84]

Supported protocol capability USB3, 32-bit dword #1: Name String "USB "

Figure 3-66. SUPTPRT3_DW1 Register

31	24	23	16	15	8	7	0
CHAR3		CHAR2		CHAR1		CHAR0	
R-0x20		R-0x42		R-0x53		R-0x55	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-74. SUPTPRT3_DW1 Register Field Descriptions

Bit	Name	Description
31-24	CHAR3	ASCII " " (space)
23-16	CHAR2	ASCII "B"
15-8	CHAR1	ASCII "S"
7-0	CHAR0	ASCII "U"

3.2.7.8 SUPTPRT3_DW2 [Offset = 0x10A88]

Supported protocol capability USB3, 32-bit dword #2

Figure 3-67. SUPTPRT3_DW2 Register

31	28	27	17	16	15	8	7	0
PSIC		Reserved		Reserved	CPC		CPO	
R-0		R-0		R-0	R-1		R-2	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-75. SUPTPRT3_DW2 Register Field Descriptions

Bit	Name	Description
31-28	PSIC	Port Speed ID Count. Reserved in xHCI 0.96 • SS (0) = USB3 Super-Speed supported

Table 3-75. SUPTPRT3_DW2 Register Field Descriptions (continued)

Bit	Name	Description
27-17	Reserved	
16	Reserved	
15-8	CPC	Compatible Port Count: Number of consecutive ports of the root hub that support this protocol, from CPO to CPO+CPC-1
7-0	CPO	Compatible Port Offset: Starting port number of root hub port(s) that support this protocol.

3.2.8 Global Registers[Offset = 0x1C100]

Table 3-76. Global Registers

Offset	Acronym	Register Description	Section
0x1C100	GSBUSCFG0	Global SoC Bus Configuration register 0	Section 3.2.8.1
0x1C104	GSBUSCFG1	Global SoC Bus Configuration register 1	Section 3.2.8.2
0x1C108	GTXTHRCFG	Global Tx Threshold Control register (valid only in Host mode)	Section 3.2.8.3
0x1C10C	GRXTHRCFG	Global Rx Threshold Control register (valid only in Host mode)	Section 3.2.8.4
0x1C110	GCTL	Global Control register	Section 3.2.8.5
0x1C118	GSTS	Global Status register	Section 3.2.8.6
0x1C120	GCOREID	USB Core Identification and release number register	Section 3.2.8.7
0x1C124	GGPIO	Global General Purpose Input/Output register	Section 3.2.8.8
0x1C128	GUID	Global User ID register	Section 3.2.8.9
0x1C12C	GUCTL	Global User Control register	Section 3.2.8.10
0x1C130	GBUSERRADDRLO	Global Bus Error Address register, lower half	Section 3.2.8.11
0x1C134	GBUSERRADDRHI	Global Bus Error Address register, upper half	Section 3.2.8.12
0x1C138	GPRTBIMAPLO	Global Port-to-SS USB Instance Mapping register, lower half	Section 3.2.8.13
0x1C13C	GPRTBIMAPHI	Global Port-to-SS USB Instance Mapping register, upper half	Section 3.2.8.14
0x1C140	GHWPARAMS0	Global Hardware Parameters register 0	Section 3.2.8.15
0x1C144	GHWPARAMS1	Global Hardware Parameters register 1	Section 3.2.8.16
0x1C148	GHWPARAMS2	Global Hardware Parameters register 2	Section 3.2.8.17
0x1C14C	GHWPARAMS3	Global Hardware Parameters register 3	Section 3.2.8.18
0x1C150	GHWPARAMS4	Global Hardware Parameters register 4	Section 3.2.8.19
0x1C154	GHWPARAMS5	Global Hardware Parameters register 5	Section 3.2.8.20
0x1C158	GHWPARAMS6	Global Hardware Parameters register 6	Section 3.2.8.21
0x1C15C	GHWPARAMS7	Global Hardware Parameters register 7	Section 3.2.8.22
0x1C180	GPRTBIMAP_HS_LO	Global Port to USB Instance Mapping register, HS, lower bits.	Section 3.2.8.23
0x1C184	GPRTBIMAP_HS_HI	Global Port to USB Instance Mapping register, HS, upper bits.	Section 3.2.8.24
0x1C188	GPRTBIMAP_FS_LO	Global Port to USB Instance Mapping register, FS/LS, lower bits.	Section 3.2.8.25
0x1C18C	GPRTBIMAP_FS_HI	Global Port to USB Instance Mapping register, FS/LS, upper bits.	Section 3.2.8.26
0x1C200	GUSB2PHYCFG	Global USB2 PHY <UTMI and ULPI (not support on K2 device)> Configuration register	Section 3.2.8.27
0x1C280	GUSB2PHYACC	Global USB2 PHY Access register	Section 3.2.8.28
0x1C2C0	GUSB3PIPECTL	Global USB3 PIPE Control register	Section 3.2.8.29
0x1C300+4*N	GTXFIFOSIZ[N]	Global Tx FIFO Size array (starting address and depth in RAM1)	Section 3.2.8.30

Table 3-76. Global Registers (continued)

Offset	Acronym	Register Description	Section
0x1C380+4*N	GRXFIFOSIZ[N]	Global Rx FIFO Size array (starting address and depth in RAM0)	Section 3.2.8.31.1
0x1C400+16*N	GEVNTADR_LO[N]	Global Event buffer Address address, lower bits	Section 3.2.8.32
0x1C404+16*N	GEVNTADR_HI[N]	Global Event buffer Address address, upper bits	Section 3.2.8.33
0x1C408+16*N	GEVNTSIZ[N]	Global Event buffer Size array	Section 3.2.8.34
0x1C40C+16*N	GEVNTCOUNT[N]	Global Event Count array	Section 3.2.8.35
0x1C600	GHWPARAMS8	Global Hardware Parameters register 8	Section 3.2.8.36

3.2.8.1 GSBUSCFG0 [Offset = 0x1C100]

Global SoC Bus Configuration Register 0

Figure 3-68. GSBUSCFG0 Register

31 13	12	11	10 8	7	6
Reserved	DESCBIGEND	DATBIGEND	Reserved	INCR256BRSTENA	INCR128BRSTENA
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

5	4
INCR64BRSTENA	INCR32BRSTENA
R/W-0	R/W-0

3	2	1	0
INCR16BRSTENA	INCR8BRSTENA	INCR4BRSTENA	INCRBRSTENA
R/W-1	R/W-1	R/W-1	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-77. GSBUSCFG0 Register Field Descriptions

Bit	Name	Description
31-13	Reserved	
12	DESCBIGEND	Endian mode for descriptor accesses. • LITTLE (0) = Little-Endian (default) • BIG (1) = Big-Endian
11	DATBIGEND	Endian mode for data accesses. • LITTLE (0) = Little-Endian (default) • BIG (1) = Big-Endian
10-8	Reserved	
7	INCR256BRSTENA	INCR256 Burst Type Enable. 256*64/8= 2-kByte burst.
6	INCR128BRSTENA	INCR128 Burst Type Enable. 128*64/8= 1-kByte burst.
5	INCR64BRSTENA	INCR64 Burst Type Enable. 64*64/8= 512-Byte burst.
4	INCR32BRSTENA	INCR32 Burst Type Enable. 32*64/8= 256-Byte burst.
3	INCR16BRSTENA	INCR16 Burst Type Enable. 16*64/8= 128-Byte burst.
2	INCR8BRSTENA	INCR8 Burst Type Enable. 8*64/8= 64-Byte burst.
1	INCR4BRSTENA	INCR4 Burst Type Enable. 4*64/8= 32-Byte burst: RECOMMENDED Enables bursts of beat length 1, 2, 3, 4, and prevents (16-byte) descriptor accesses from being broken up: highly recommended.

Table 3-77. GSBUSCFG0 Register Field Descriptions (continued)

Bit	Name	Description
0	INCRBRSTENA	Undefined Length INCR Burst Type Enable: DO NOT ENABLE When enabled, this has higher priority than other burst types.

3.2.8.2 GSBUSCFG1 [Offset = 0x1C104]

Global SoC Bus Configuration Register 1

Figure 3-69. GSBUSCFG1 Register

31 13	12	11 8	7 0
Reserved	EN1KPAGE	PIPETRANSLIMIT	Reserved
R-0	R/W-0	R/W- 0xF	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-78. GSBUSCFG1 Register Field Descriptions

Bit	Name	Description
31-13	Reserved	
12	EN1KPAGE	1k-page Boundary Enable • DIS (0) = Master breaks requests only on 4kbyte boundaries • EN (1) = Master breaks requests on all 1kbyte boundaries
11-8	PIPETRANSLIMIT	Maximum number of outstanding (read or write) pipelined sequential (i.e. in-order) transaction requests on the master interface (field value+1) • _1 (0) = Single request mode • _2 (1) = up to 2 pending requests • _16 (15) = up to 16 pending requests (maximum)
7-0	Reserved	Always write 0, result of writing of a non-zero value is undetermined.

3.2.8.3 GTXTHRCFG [Offset = 0x1C108]

Global Tx Threshold Control Register. Valid only in Host mode.

Figure 3-70. GTXTHRCFG Register

31 30	29	28	27 24
Reserved	USBTXPKTCNTSEL	Reserved	USBTXPKTCNT
R/W-0	R/W-0	R-0	R/W-0

23 16	15 14	13 11	10 0
USBMAXTXBURSTSIZE	Reserved	Reserved	Reserved
R/W-0	R/W-0	R-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-79. GTXTHRCFG Register Field Descriptions

Bit	Name	Description
31-30	Reserved	Always write 0, result of writing of a non-zero value is undetermined.
29	USBTXPKTCNTSEL	USB Transmit Packet Count Enable: Enables/disables USB trasnmission multi-packet thresholding • DIS (0) = Disabled: Transmission is only started on the USB after the entire packet has been fetched into the corresponding TXFIFO. Rest of the register must be all-zero. • EN (1) = Enabled: USB transmission is started only after USB Transmit Packet Count amount of packets for the USB transaction (burst) are already in the corresponding TXFIFO This mode is only valid in the host, superspeed mode.
28	Reserved	

Table 3-79. GTXTHRCFG Register Field Descriptions (continued)

Bit	Name	Description
27-24	USBTXPKTCNT	USB Transmit Packet Count : Number of packets that must be in the TXFIFO before transmission for the corresponding USB transaction (burst) can start. Don't care if USBTxPktCntSel=0. • DIS (0) = Use when and only when disabled (USBTxPktCntSel=0) • MIN (1) = Minimum value (when enabled) • MAX (15) = Maximum value
23-16	USBMAXTXBURSTSIZE	USB Maximum Transmit Burst Size. Max OUT burst size, when USBTxPktCntSel=1. Avoids TX FIFO underrun when the system bus is slower than the USB. Only applies to SS Bulk / Iso / Int OUT endpoints in host mode. Don't care if USBTxPktCntSel=0. • MIN (1) = Burst of up to 1 (minimum value when enabled) • MAX (16) = Burst of up to 16 (maximum value) • DIS (0) = Use when and only when disabled (USBTxPktCntSel=0)
15-14	Reserved	Always write 0, result of writing of a non-zero value is undetermined.
13-11	Reserved	
10-0	Reserved	Always write 0, result of writing of a non-zero value is undetermined.

3.2.8.4 GRXTHRCFG [Offset = 0x1C10C]

Global Rx Threshold Control Register. Valid only in Host mode.

Figure 3-71. GRXTHRCFG Register

31	30	29	28	27	24
Reserved		USBRXPKTCNTSEL	Reserved	USBRXPKTCNT	
R-0		R/W-0	R-0	R/W-0	

23	19	18	16	15	14	11	10	0
USBMAXRXBURSTSIZE		Reserved		Reserved	Reserved		Reserved	
R/W-0		R-0		R/W-0	R-0		R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-80. GRXTHRCFG Register Field Descriptions

Bit	Name	Description
31-30	Reserved	
29	USBRXPKTCNTSEL	USB ReceivePacket Count Enable Enables/disables USB reception multi-packet thresholding • DIS (0) = The core can only start reception on the USB when the RX FIFO has space for at least one packet. Rest of the register must be all-zero. • EN (1) = The core can only start reception on the USB when the RX FIFO has space for at least packet. This mode is only valid in the host mode. It is only used for SuperSpeed.
28	Reserved	
27-24	USBRXPKTCNT	USB Receive Packet Count: Number of packets that must be available in the RX FIFO before the core can start the corresponding USB RX transaction (burst). Don't care if USBRxPktCntSel=0. • DIS (0) = Use when and only when disabled (USBRxPktCntSel=0) • MIN (1) = Minimum value (when enabled) • MAX (15) = Maximum value

Table 3-80. GRXTHRCFG Register Field Descriptions (continued)

Bit	Name	Description
23-19	USBMAXRXBURSTSIZE	USB Maximum Receive Burst Size. Maxi IN burst size, when USBRxPktCntSel = 1. When the system bus is slower than the USB, RX FIFO can overrun during a long burst. User can program a smaller value to this field to limit the RX burst size that the core can do. Only applies to SS Bulk / Iso / Int IN endpoints in host mode. Don't care if USBRxPktCntSel=0. • MIN (1) = Burst of up to 1 (minimum value when enabled) • MAX (16) = Burst of up to 16 (maximum value) • DIS (0) = Use when and only when disabled (USBRxPktCntSel=0)
18-16	Reserved	
15	Reserved	Always write 0, result of writing of a non-zero value is undetermined.
14-11	Reserved	
10-0	Reserved	Always write 0, result of writing of a non-zero value is undetermined.

3.2.8.5 GCTL [Offset = 0x1C110]

Global Control Register

Figure 3-72. GCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12
PWRDNSCALE													MASTERFILTBYPASS	BYPSSETADDR	U2RSTECN	FRMSCLDWN		PRTCAPDIR	
R/W-0x04B0													R/W-0	R/W-0	R/W-0	R/W-0		R/W-0x3	

11	10	9	8
CORESOFTRESET	Reserved		DEBUGATTACH
R/W-0	R-0		R/W-0

7	6	5	4	3	2	1	0
RAMCLKSEL		SCALEDOWN		DISSCRAMBLE	Reserved		DSBLCLKGTNG
R/W-0		R/W-0		R/W-0	R-0		R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-81. GCTL Field Descriptions

Bit	Name	Description
31-19	PWRDNSCALE	Power Down Scale: In P3 state, pipe clock stops and is replaced internally by the suspend clock to create a 16kHz reference. Set field to Fs/16k, rounded up, with Fp suspend clock frequency. Required accuracy is 0-50% • _32K_MIN (2) = 32kHz suspend clock (minimum frequency) • _12M (750) = 12 MHz suspend clock • _13M (813) = 13 MHz suspend clock • _19M2 (1200) = 19.2 MHz suspend clock • _24M (1500) = 24 MHz suspend clock • _26M (1625) = 26 MHz suspend clock • _38M4 (2400) = 38.4 MHz suspend clock • _125M (7813) = 125 MHz suspend clock
18	MASTERFILTBYPASS	Master Filter Bypass. Bypasses the double-synchronizers and the 5 ms debounce filters on UTMI+ inputs (the latter are not implemented).
17	BYPSSETADDR	Override of the device address, bypassing the SET ADDRESS control transfer. For simulation only. • FUNC (0) = Functional mode • BYPASS (1) = Device's address set directly to DCFG.DevAddr
16	U2RSTECN	If the super speed connection fails during POLL or LMP exchange, the device connects at non-SS mode. If this bit is set, then device attempts three more times to connect at SS, even if it previously failed to operate in SS mode.

Table 3-81. GCTL Field Descriptions (continued)

Bit	Name	Description
15-14	FRMSCLDWN	Frame scale-down This field scales down device view of a SOF (FS/LS) / uSOF (HS) / ITP (SS) duration. • _0 (0) = SS/HS interval=125 us, FS interval=1ms (standard) • _1 (1) = SS/HS interval=62.5 us, FS interval=0.5ms • _2 (2) = SS/HS interval=31.25 us, FS interval=0.25ms • _3 (3) = SS/HS interval=15.625 us, FS interval=0.125ms
13-12	PRTCAPDIR	Port Capability Direction • HST (1) = Core acts as DRD in host mode • DEV (2) = Core acts as DRD in device mode • DRD (3) = Core acts as OTG DRD, with the mode set by the ID
11	CORESOFTRESET	Core Soft Reset. When you reset PHYs (using GUBS3PHYCFG or GUSB3PIPECTL registers), you must keep the core in reset state until PHY clocks are stable. • NO (0) = No soft reset • RESET (1) = Soft reset to core
10-9	Reserved	
8	DEBUGATTACH	Debug Attach. When this bit is set: a) SS Link proceeds directly to the Polling link state (after RUN/STOP in the DCTL register is asserted) without checking remote termination. b) Link LFPS polling timeout is infinite c) Polling timeout during TS1 is infinite (in case link is waiting for TXEQ to finish).
7-6	RAMCLKSEL	RAM Clock Select. No action, hardware always uses bus clock (config 2'b00) • BUS (0) = AXI bus clock • PIPE (1) = PIPE clock • PIPE_50 (2) = PIPE/2 clock • MAC (3) = MAC clock (125 MHz)
5-4	SCALEDOWN	Scale-Down Mode Enable Switches to shorter, non-standard protocol time intervals to speed up simulation. DO NOT MODIFY ON ACTUAL HARDWARE. • NONE (0) = Disable scale-downs. Legal USB timing values are used. Always use on actual hardware. • _1 (1) = Enables scaled down values of : - HS/FS/LS: all except Device-mode suspend and resume. - SS: Number of TxEq training sequences reduce to 8 - SS: LFPS polling burst time reduce to 100 ns - SS: LFPS warm reset receive reduce to 30 us - SS: more ... • _2 (2) = HS/FS/LS: scale-down of Device-mode suspend and resume. SS: No TxEq training sequences are sent. • _3 (3) = SS/HS/FS/LS: scale down all the above
3	DISSCRAMBLE	Disable Scrambling. Transmit request to Link Partner on next transition to Recovery or Polling.
2-1	Reserved	
0	DSBLCLKGTNG	Disable Clock Gating. When this bit is set to 1 and the core is in Low Power mode, internal clock gating is disabled.

3.2.8.6 GSTS [Offset = 0x1C118]

Global Status Register

Figure 3-73. GSTS Register

31	20	19	11	10	9	8
CBELT		Reserved		OTG_IP	BC_IP	ADP_IP
R-0x3E8		R-0		R-0	R-0	R-0

7	6	5	4	3	2	1	0
HOST_IP	DEVICE_IP	CSRTIMEOUT	BUSERRADDRVLD		Reserved		CURMOD
R-0	R-0	R/W-0	R/W-0		R-0		R-0x2

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-82. GSTS Register Field Descriptions

Bit	Name	Description
31-20	CBELT	Current BELT Value In Host mode, this field indicates the minimum value of all received device BELT values and the BELT value that is set by the Set Latency Tolerance Value command.
19-11	Reserved	
10	OTG_IP	OTG interrupt status • NONE (0) = No interrupt pending • PEND (1) = Interrupt pending
9	BC_IP	Battery Charger interrupt status: NOT IMPLEMENTED
8	ADP_IP	ADP interrupt status: NOT IMPLEMENTED
7	HOST_IP	Host interrupt status • NONE (0) = No interrupt pending • PEND (1) = Interrupt pending
6	DEVICE_IP	Device interrupt status • NONE (0) = No interrupt pending • PEND (1) = Interrupt pending
5	CSRTIMEOUT	Control/Status Register access Timeout status flag. • NOEVENT (0) = no CSR timeout • SET (1) = CSR access timed out after 65,535 clock cycles • CLEAR (1) = Clear the status flag. • NOACTION (0) = No action
4	BUSERRADDRVLD	Bus Error Address Valid status flag. Also flagged on USBSTS.HSE field (host mode) and DEPEVT[12] on XferComplete/XferInProgress event (device mode). • NOEVENT (0) = No bus error since last clear; GBUSERRADDR is not valid • SET (1) = At least one bus error took place, the first one near address GBUSERRADDR. • CLEAR (1) = Clear the status flag. • NOACTION (0) = No action
3-2	Reserved	
1-0	CURMOD	Current Mode of Operation. • DEV (0) = Device • HOST (1) = Host • DRD (2) = DRD

3.2.8.7 GCOREID [Offset = 0x1C120]

Core ID: Core Identification and Release number.

Software uses this register to configure release-specific features in the driver.

Figure 3-74. GCOREID Register

31	16	15	0
COREID_CORE		COREID_REL	
R-0x5533		R-0x180A	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-83. GCOREID Register Field Descriptions

Bit	Name	Description
31-16	COREID_CORE	COREID MSBytes: core identifier • ID (21811) = ASCII for "U3" = USB3
15-0	COREID_REL	COREID LSBytes: version number For instance, version 1.00a => 0x100A • _1_83A (6202) = version 1.83a • _2_02A (8234) = version 2.02a

3.2.8.8 GGPIO [Offset = 0x1C124]

Global General Purpose Input/Output Register

Figure 3-75. GGPIO Register

31	16	15	0
GPO		GPI	
R/W-0		R-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-84. GGPIO Register Field Descriptions

Bit	Name	Description
31-16	GPO	General Purpose Output. DO NOT USE: NOT CONNECTED.
15-0	GPI	General Purpose Inputs. TIED LOW.

3.2.8.9 GUID [Offset = 0x1C128]

Global User ID Register

Figure 3-76. GUID Register

31	0
USERID	
R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-85. GUID Register Field Descriptions

Bit	Name	Description
31-0	USERID	Application-programmable ID field.

3.2.8.10 GUCTL [Offset = 0x1C12C]

Global User Control Register

Figure 3-77. GUCTL Register

31	22	21	20	18	17	16	15	14	13	11	10	9	8	0
Reserved		NOEXTRDL	PSQEXTRRESSP		SPRSCTRLTRANSEN	RESBWHSEPS	CMDEVADDR	USBHSTINAUTORETRYEN	Reserved		DTCT		DTFT	
R/W-0x7FC0		R/W-0	R/W-0		R/W-0	R/W-0	R/W-1	R/W-0	R/W-0		R/W-0		R/W-0x10	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-86. GUCTL Register Field Descriptions

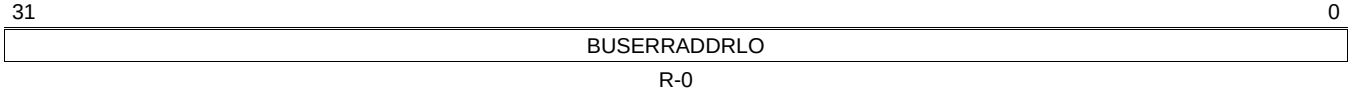
Bit	Name	Description
31-22	Reserved	
21	NOEXTRDL	No Extra Delay between SOF and the 1st packet (when host) • DIS (0) = 2 us delay between SOF and first packet. Some bandwith is lost. • EN (1) = No extra delay. May cause some devices to misbehave.
20-18	PSQEXTRRESSP	Protocol Status Queue Extra Reserved Space (Debug only). Additional space in the PSQ reserved before the USB3 protocl transaction layer (U3PTL) initiates a new USB transaction and burst beats. • DIS (0) = • EN (1) =
17	SPRSCTRLTRANSEN	Sparse Control Transaction Enable. Valid in host mode only (any speed). • DIS (0) = Feature disabled, no limitation on control transactions • EN (1) = Host controller schedules transactions for Control transfer in different [micro]frames to prevent incorrect behaviour from device.
16	RESBWHSEPS	Reserving (more) Bandwidth for HS Periodic EPs. Valid in host mode only. • _80 (0) = HC reserves 80% of the bandwidth for periodic EPs • _85 (1) = HC reserves 85% of the bandwidth for periodic EPs, which allows 2 high speed, high bandwidth ISOC EPs.
15	CMDEVADDR	Compliance Mode for Device Address. Valid in host mode only. • EQ (0) = Device Address is equal to Slot ID. • DIFF (1) = Increment Device Address on each Address Device command. Slot ID may have different value than Device Address if max_slot_enabled < 128.
14	USBHSTINAUTORETRYEN	Host IN Auto Retry Enable: host core behaviour upon data packet CRC errors or internal overrun scenarios in non-isochronous IN transfers. • DIS (0) = Auto Retry Disabled: Host core to reply with a terminating retry ACK (Retry=1 and NumP=0) • EN (1) = Auto Retry Enabled: Host core to reply with a non-terminating retry ACK (Retry=1 and NumP!=0)
13-11	Reserved	
10-9	DTCT	Device Timeout Coarse Tuning: time the host waits for a response from device before timeout. Coarse setting. • FINE (0) = Use DTFT instead • _0M5 (1) = 500 us • _1M5 (2) = 1.5 ms • _5MS (3) = 6.5 ms
8-0	DTFT	Device Timeout Fine Tuning: time the host waits for a response from device before timeout. Fine setting. Timer runs on the 125 MHz clock (8 ns period), timeout is DTFT * 256 * 8 ns ~= DTFT * 2 us Don't care unless DTCT=0 • _2US (1) = ~ 2 us timeout (2,048 ns) • _10US (5) = ~ 10 us timeout (10,240 ns) • _100US (50) = ~ 100 us timeout (102,400 ns) • _256US (125) = 256 us timeout (maximum, exact value)

3.2.8.11 GBUSERRADDRLO [Offset = 0x1C130]

Global Bus Error (non-precise) Address, LSbits: Base address of the first system bus DMA transfer that got a bus error. Note that each DMA transfer can contain several bursts, each spanning several addresses.

Valid when GSTS.BusErrAddrVld=1. Cleared upon core reset.

Figure 3-78. GBUSERRADDRLO Register



Legend: R = Read only; W = Write only; -n = value after reset

Table 3-87. GBUSERRADDRLO Register Field Descriptions

Bit	Name	Description
31-0	BUSERRADDRLO	BUSERRADDR[31:0]

3.2.8.12 GBUSERRADDRHI [Offset = 0x1C134]

Global Bus Error (non-precise) Address, MSbits: Base address of the first system bus DMA transfer that got a bus error. Note that each DMA transfer can contain several bursts, each spanning several addresses.

Valid when GSTS.BusErrAddrVld=1. Cleared upon core reset.

Figure 3-79. GBUSERRADDRHI Register

31	0
BUSERRADDRHI	
R-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-88. GBUSERRADDRHI Register Field Descriptions

Bit	Name	Description
31-0	BUSERRADDRHI	BUSERRADDR[63:32]

3.2.8.13 GPRTBIMAPLO [Offset = 0x1C138]

Global Port-to-SS USB Instance Mapping, low bits [31:0]

Figure 3-80. GPRTBIMAPLO Register

31	4	3	0
Reserved		BINUM1	
R-0		R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-89. GPRTBIMAPLO Register Field Descriptions

Bit	Name	Description
31-4	Reserved	
3-0	BINUM1	SS USB Instance Number for Port number 1 Application-programmable ID field.

3.2.8.14 GPRTBIMAPHI [Offset = 0x1C13C]

Global Port-to-SS USB Instance Mapping, high bits [63:32]

Figure 3-81. GPRTBIMAPHI Register

31	0
Reserved	
R-0	

Legend: R = Read only; W = Write only; -n = value after reset

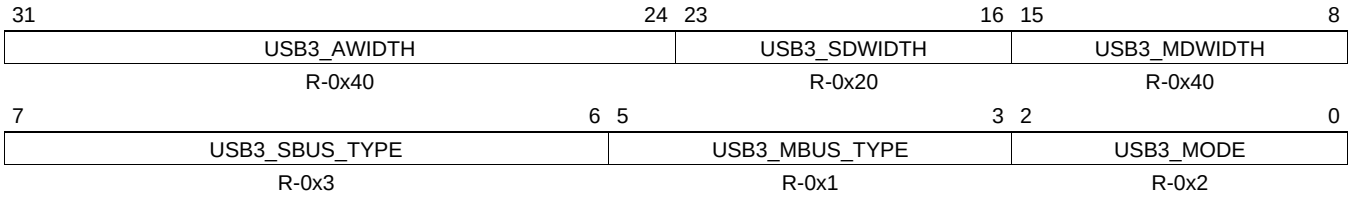
Table 3-90. GPRTBIMAPHI Register Field Descriptions

Bit	Name	Description
31-0	Reserved	

3.2.8.15 GHWPARAMS0 [Offset = 0x1C140]

Global hardware parameters #0

Figure 3-82. GHWPARAMS0 Register



Legend: R = Read only; W = Write only; -n = value after reset

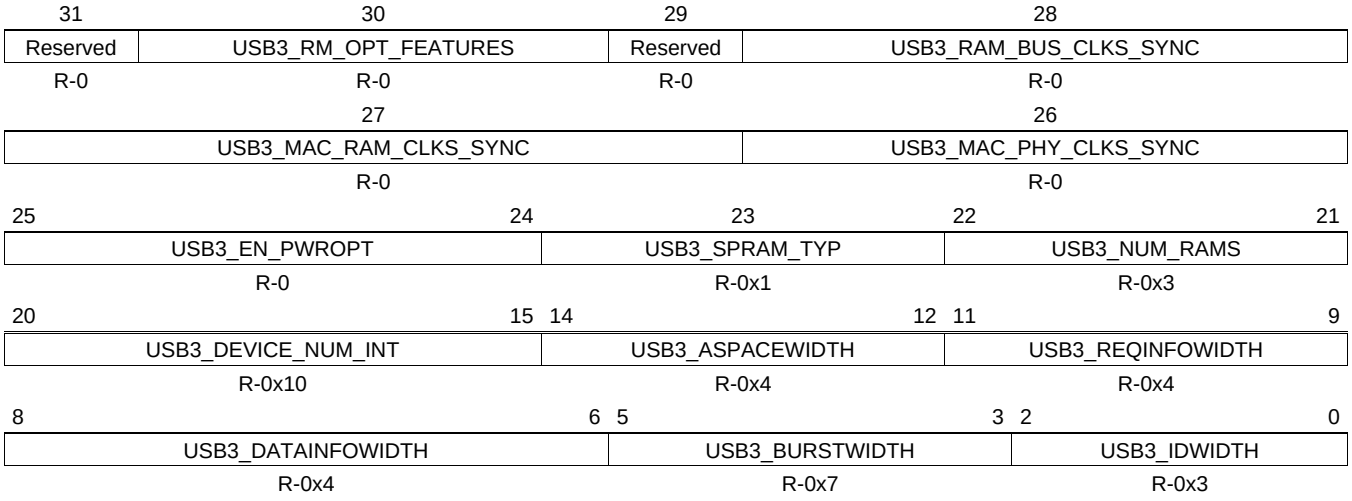
Table 3-91. GHWPARAMS0 Register Field Descriptions

Bit	Name	Description
31-24	USB3_AWIDTH	Global hardware configuration parameter USB3_AWIDTH: (Master) Address Width (in bits)
23-16	USB3_SDWIDTH	Global hardware configuration parameter USB3_SDWIDTH: Slave Data Width (in bits)
15-8	USB3_MDWIDTH	Global hardware configuration parameter USB3_MDWIDTH: Master Data Width (in bits)
7-6	USB3_SBUS_TYPE	Global hardware configuration parameter USB3_SBUS_TYPE: (System bus) Slave type • NATIVE (0) = Native slave
5-3	USB3_MBUS_TYPE	Global hardware configuration parameter USB3_MBUS_TYPE: (System bus) Master type • AXI (1) = AXI (AMBA 3) master system bus
2-0	USB3_MODE	Global hardware configuration parameter USB3_MODE • DEV (0) = Device-only • HST (1) = Host-only • DRD (2) = Dual-role device (DRD) i.e. both device and host

3.2.8.16 GHWPARAMS1 [Offset = 0x1C144]

Global hardware parameters #1

Figure 3-83. GHWPARAMS1 Register



Legend: R = Read only; W = Write only; -n = value after reset

Table 3-92. GHWPARAMS1 Register Field Descriptions

Bit	Name	Description
31	Reserved	
30	USB3_RM_OPT_FEATURES	Global hardware configuration parameter USB3_RM_OPT_FEATURES: Remove Optional Features • NO (0) = • YES (1) =
29	Reserved	
28	USB3_RAM_BUS_CLKS_SYNC	Global hardware configuration parameter USB3_RAM_BUS_CLKS_SYNC: RAM vs. BUS clocks synchronous ? • NO (0) = • YES (1) =
27	USB3_MAC_RAM_CLKS_SYNC	Global hardware configuration parameter USB3_MAC_RAM_CLKS_SYNC: MAC vs. RAM clocks synchronous ? • NO (0) = • YES (1) =
26	USB3_MAC_PHY_CLKS_SYNC	Global hardware configuration parameter USB3_MAC_PHY_CLKS_SYNC: MAC vs. PHY clocks synchronous ? • NO (0) = • YES (1) =
25-24	USB3_EN_PWROPT	Global hardware configuration parameter USB3_EN_PWROPT: Power optimization • NONE (0) = • CLOCK (1) = • CLOCK_HIBERNATION (2) =
23	USB3_SPRAM_TYP	Global hardware configuration parameter USB3_SPRAM_TYP • SP (1) = Single-port RAM
22-21	USB3_NUM_RAMS	Global hardware configuration parameter USB3_NUM_RAMS: Number of internal RAMs • _1 (1) = Single-RAM • _2 (2) = Two-RAM • _3 (3) = Three-RAM
20-15	USB3_DEVICE_NUM_INT	Global hardware configuration parameter USB3_DEVICE_NUM_INT: Number of interrupts (and event buffers) in device mode
14-12	USB3_ASPACEWIDTH	Global hardware configuration parameter USB3_ASPACEWIDTH
11-9	USB3_REQINFOWIDTH	Global hardware configuration parameter USB3_REQINFOWIDTH
8-6	USB3_DATAINFOWIDTH	Global hardware configuration parameter USB3_DATAINFOWIDTH
5-3	USB3_BURSTWIDTH	Global hardware configuration parameter USB3_BURSTWIDTH minus one, fixed to 8-1=7
2-0	USB3_IDWIDTH	Global hardware configuration parameter USB3_IDWIDTH minus 1 Note: Sets only the master port's ID width. Slave ID width is set by non-readable USB3_SIDWIDTH

3.2.8.17 GHWPARAMS2 [Offset = 0x1C148]

Global hardware parameters #2

Figure 3-84. GHWPARAMS2 Register

31	0
USB3_USERID	
R-0	

Legend: R = Read only; W = Write only; -n = value after reset

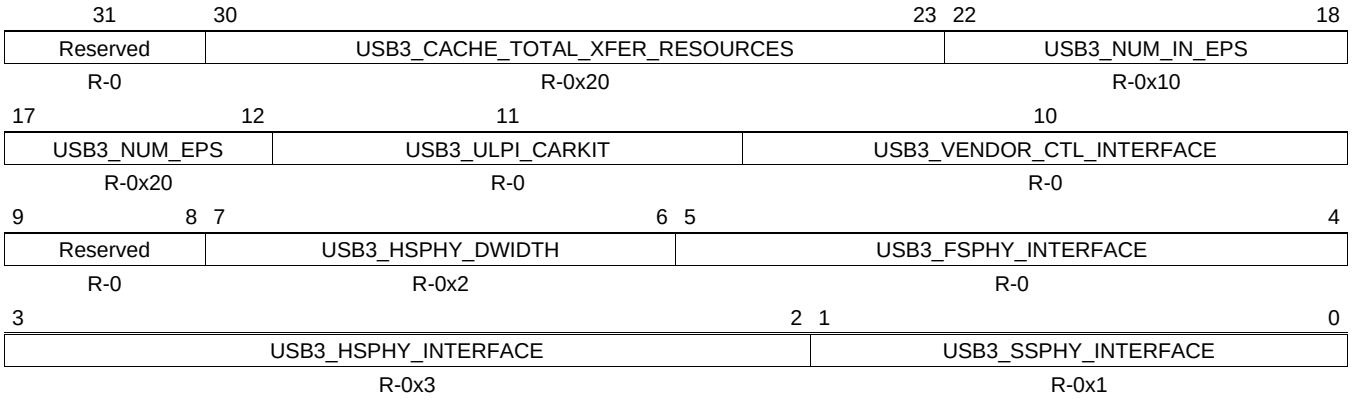
Table 3-93. GHWPARAMS2 Register Field Descriptions

Bit	Name	Description
31-0	USB3_USERID	Global hardware configuration parameter USB3_USERID

3.2.8.18 GHWPARAMS3 [Offset = 0x1C14C]

Global hardware parameters #3

Figure 3-85. GHWPARAMS3 Register



Legend: R = Read only; W = Write only; -n = value after reset

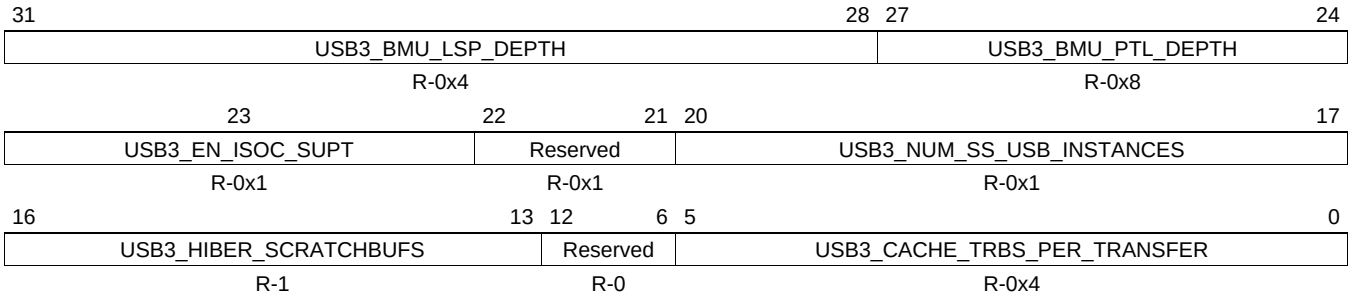
Table 3-94. GHWPARAMS3 Register Field Descriptions

Bit	Name	Description
31	Reserved	
30-23	USB3_CACHE_TOTAL_XFER_RESOURCES	Global hardware configuration parameter USB3_NUM_CACHE_TOTAL_XFER_RESOURCES: Cache total transfer resources
22-18	USB3_NUM_IN_EPS	Global hardware configuration parameter USB3_NUM_IN_EPS: Number of IN endpoints, with EP0 counting as one.
17-12	USB3_NUM_EPS	Global hardware configuration parameter USB3_NUM_EPS: Total number of endpoints (IN+OUT, with EP0 counting as 2 separate ones)
11	USB3_ULPI_CARKIT	Global hardware configuration parameter USB3_ULPI_CARKIT: ULPI (optional) car-kit mode implementation • NO (0) = CK mode not implemented • VC (1) = CK mode implemented
10	USB3_VENDOR_CTL_INTERFACE	Global hardware configuration parameter USB3_VENDOR_CTL_INTERFACE: (UTMI) Vendor Control i/f implementation • NO (0) = VC i/f not implemented • VC (1) = VC i/f implemented
9-8	Reserved	
7-6	USB3_HSPHY_DWIDTH	Global hardware configuration parameter USB3_HSPHY_DWIDTH: HS PHY data width • _8 (0) = 8-bit UTMI @60 MHz • _16 (1) = 16-bit UTMI @30 MHz • _8_16 (2) = 8/16 bit UTMI @60/30 MHz
5-4	USB3_FSPHY_INTERFACE	Global hardware configuration parameter USB3_FSPHY_INTERFACE: Full (/Low)-Speed (serial) PHY interface • NONE (0) = No FS/LS PHY i/f
3-2	USB3_HSPHY_INTERFACE	Global hardware configuration parameter USB3_HSPHY_INTERFACE: High-speed PHY interface • NONE (0) = No HS (/FS/LS) PHY i/f • UTMI (1) = UTMI+ i/f • ULPI (2) = ULPI i/f • BOTH (3) = UTMI+ and ULPI i/f (mutually exclusive operation)
1-0	USB3_SSPHY_INTERFACE	Global hardware configuration parameter USB3_SSPHY_INTERFACE: Super Speed PHY interface. • NONE (0) = No SS PHY i/f • PIPE (1) = PIPE i/f for USB3.0

3.2.8.19 GHWPARAMS4 [Offset = 0x1C150]

Global hardware parameters #4

Figure 3-86. GHWPARAMS4 Register



Legend: R = Read only; W = Write only; -n = value after reset

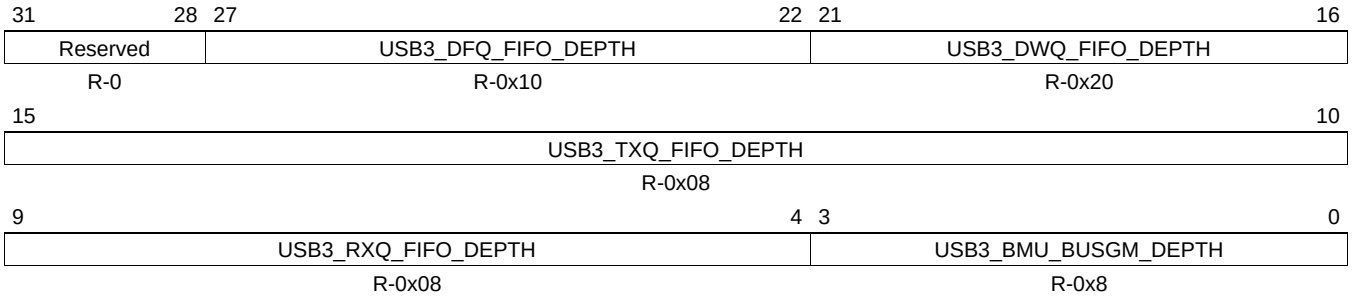
Table 3-95. GHWPARAMS4 Register Field Descriptions

Bit	Name	Description
31-28	USB3_BMU_LSP_DEPTH	Global hardware configuration parameter USB3_BMU_LSP_DEPTH: Bus Management Unit / List Processor buffer depth
27-24	USB3_BMU_PTL_DEPTH	Global hardware configuration parameter USB3_BMU_PTL_DEPTH: Bus Management Unit / Protocol Transaction Layer buffer depth
23	USB3_EN_ISOC_SUPT	Global hardware configuration parameter USB3_EN_ISOC_SUPT: Enable Isochronous Support • NONE (0) = Isochronous not supported • ISO (1) = Isochronous transfer type supported
22-21	Reserved	
20-17	USB3_NUM_SS_USB_INSTANCES	Global hardware configuration parameter USB3_NUM_SS_USB_INSTANCES: Number of (independent) SS USB schedulers
16-13	USB3_HIBER_SCRATCHBUFS	Global hardware configuration parameter USB3_HIBER_SCRATCHBUFS: Number of 4kbyte buffers required in system memory to store context during hibernation. Don't care since hibernation is not enabled.
12-6	Reserved	
5-0	USB3_CACHE_TRBS_PER_TRANSFER	Global hardware configuration parameter USB3_CACHE_TRBS_PER_TRANSFER

3.2.8.20 GHWPARAMS5 [Offset = 0x1C154]

Global hardware parameters #5

Figure 3-87. GHWPARAMS5 Register



Legend: R = Read only; W = Write only; -n = value after reset

Table 3-96. GHWPARAMS5 Register Field Descriptions

Bit	Name	Description
31-28	Reserved	
27-22	USB3_DFQ_FIFO_DEPTH	Global hardware configuration parameter USB3_DFQ_FIFO_DEPTH
21-16	USB3_DWQ_FIFO_DEPTH	Global hardware configuration parameter USB3_DWQ_FIFO_DEPTH
15-10	USB3_TXQ_FIFO_DEPTH	Global hardware configuration parameter USB3_TXQ_FIFO_DEPTH
9-4	USB3_RXQ_FIFO_DEPTH	Global hardware configuration parameter USB3_RXQ_FIFO_DEPTH
3-0	USB3_BMU_BUSGM_DEPTH	Global hardware configuration parameter USB3_BMU_BUSGM_DEPTH

3.2.8.21 GHWPARAMS6 [Offset = 0x1C158]

Global hardware parameters #6

Figure 3-88. GHWPARAMS6 Register

31	16	15	14	13	12
USB3_RAM0_DEPTH		BUSFLTRSSUPPORT	BCSUPPORT	OTGSSSUPPORT	ADPSUPPORT
R-0xB00		R-0x1	R-0x1	R-0	R-0x1

11	10	9	8
HNPSUPPORT	SRPSUPPORT	Reserved	
R-0x1	R-0x1	R-0	

7	6	5	0
USB3_EN_FPGA	Reserved	USB3_PSQ_FIFO_DEPTH	
R-0	R-0	R-0x20	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-97. GHWPARAMS6 Register Field Descriptions

Bit	Name	Description
31-16	USB3_RAM0_DEPTH	Depth of RAM#0, in 64-bit words. RAM0 contains data cache and Rx FIFOs.
15	BUSFLTRSSUPPORT	Filtering (debounce) on OTG UTMI+ inputs (iddig,vbusvalid,avalid,bvalid,sessend). Reflects USB3_EN_OTG_FILTERS. • NO (0) = Not implemented • IMP (1) = Implemented
14	BCSUPPORT	Battery Charger detection (ACA = Accessory Charger Adapter) support implemented internally. Reflects USB3_EN_BC. Note: Support can also be provided OUTSIDE the controller. • NO (0) = ACA not supported, or supported by external logic • IMP (1) = ACA supported by internal logic
13	OTGSSSUPPORT	OTG SuperSpeed support (aka OTG3.0) • NO (0) = OTG3.0 not supported • YES (1) = OTG3.0 supported
12	ADPSUPPORT	OTG2.0 ADP (Attach Detection Protocol) support implemented internally. Reflects USB3_EN_ADP. Note: Support can also be provided OUTSIDE the controller. • NO (0) = ADP not supported, or supported by external logic • IMP (1) = ADP supported by internal logic
11	HNPSUPPORT	OTG2.0 HNP (Host Negotiation Protocol) support. Set when in DRD mode. • NO (0) = Not supported • SUPPORT (1) = Supported
10	SRPSUPPORT	OTG2.0 SRP (Session Request Protocol) support. • NO (0) = Not supported • SUPPORT (1) = Supported
9-8	Reserved	
7	USB3_EN_FPGA	Global hardware configuration parameter USB3_EN_FPGA

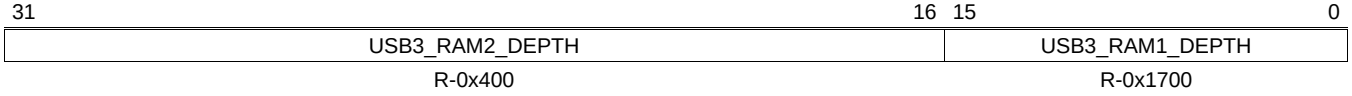
Table 3-97. GHWPARAMS6 Register Field Descriptions (continued)

Bit	Name	Description
6	Reserved	
5-0	USB3_PSQ_FIFO_DEPTH	Global hardware configuration parameter USB3_PSQ_FIFO_DEPTH

3.2.8.22 GHWPARAMS7 [Offset = 0x1C15C]

Global hardware parameters #7

Figure 3-89. GHWPARAMS7 Register



Legend: R = Read only; W = Write only; -n = value after reset

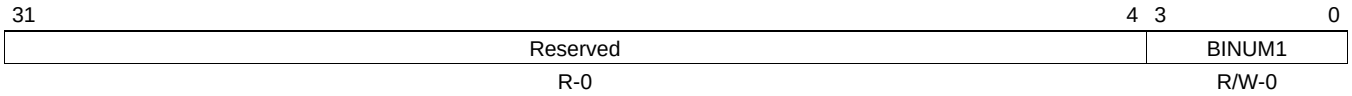
Table 3-98. GHWPARAMS7 Register Field Descriptions

Bit	Name	Description
31-16	USB3_RAM2_DEPTH	Depth of RAM#2, in 64-bit words. RAM2 IS NOT IMPLEMENTED IN 2-RAM CONFIG: don't care
15-0	USB3_RAM1_DEPTH	Depth of RAM#1, in 64-bit words. RAM1 contains Tx FIFOs.

3.2.8.23 GPRTBIMAP_HS_LO [Offset = 0x1C180]

Global Port to USB Instance Mapping Register, High-Speed, low bits [31:0]

Figure 3-90. GPRTBIMAP_HS_LO Register



Legend: R = Read only; W = Write only; -n = value after reset

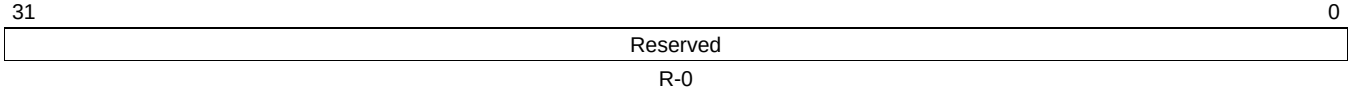
Table 3-99. GPRTBIMAP_HS_LO Register Field Descriptions

Bits	Name	Description
31-4	Reserved	
3-0	BINUM1	HS USB Instance Number for Port number 1 Application-programmable ID field.

3.2.8.24 GPRTBIMAP_HS_HI [Offset = 0x1C184]

Global Port to USB Instance Mapping Register, High-Speed, high bits [63:32]

Figure 3-91. GPRTBIMAP_HS_HI Register



Legend: R = Read only; W = Write only; -n = value after reset

Table 3-100. GPRTBIMAP_HS_HI Register Field Descriptions

Bit	Name	Description
31-0	Reserved	

3.2.8.25 GPRTBIMAP_FS_LO [Offset = 0x1C188]

Global Port to USB Instance Mapping Register, Full/low-Speed, low bits [31:0]

Figure 3-92. GPRTBIMAP_FS_LO Register

31	4	3	0
Reserved		BINUM1	
R-0		R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-101. GPRTBIMAP_FS_LO Register Field Descriptions

Bits	Name	Description
31-4	Reserved	
3-0	BINUM1	FS USB Instance Number for Port number 1 Application-programmable ID field.

3.2.8.26 GPRTBIMAP_FS_HI [Offset = 0x1C18C]

Global Port to USB Instance Mapping Register, Full/low-Speed, high bits [63:32]

Figure 3-93. GPRTBIMAP_FS_HI Register

31	0
Reserved	
R-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-102. GPRTBIMAP_FS_HI Register Field Descriptions

Bit	Name	Description
31-0	Reserved	

3.2.8.27 GUSB2PHYCFG [Offset = 0x1C200]

Global USB2 (UTMI/ULPI) PHY configuration

Figure 3-94. GUSB2PHYCFG Register

31	30	19	18	17
PHYSOFTRST	Reserved		ULPIEXTVBUSINDICATOR	ULPIEXTVBUSDRV
R/W-0	R-0		R/W-0	R/W-0

16	15	14	13	10	9
ULPICLKSUSM	ULPIAUTORES	Reserved	USBTRDTIM-0x9		Reserved
R/W-0	R/W-0	R-0	R/W-0		R-0

8	7	6	5	4	3	2	0
ENBLSLPM	PHYSEL	SUSPHY	FSINTF	ULPI_UTMI_SEL	PHYIF	TOUTCAL	
R/W-0	R-0	R/W-1	R-0	R/W-0	R/W-0	R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-103. GUSB2PHYCFG Register Field Descriptions

Bit	Name	Description
31	PHYSOFTRST	PHY Soft Reset. Active-high, fully static software reset for UTMI USB2 transceiver. • INACTIVE (0) = Reset inactive. • ACTIVE (1) = Reset active.
30-19	Reserved	

Table 3-103. GUSB2PHYCFG Register Field Descriptions (continued)

Bit	Name	Description
18	ULPIEXTVBUSINDICATOR	ULPI External VBUS Indicator Indicates the ULPI PHY VBUS over-current indicator. • INT (0) = PHY uses an internal VBUS valid comparator. • EXT (1) = PHY uses an external VBUS valid comparator.
17	ULPIEXTVBUSDRV	ULPI External VBUS Drive Selects supply source to drive 5V on VBUS, in the ULPI PHY. • INT (0) = PHY drives VBUS with internal charge pump • EXT (1) = PHY drives VBUS with an external supply.
16	ULPICLKSUSM	Sets the ClockSuspendM bit in the Interface Control register on the ULPI PHY. Applicable only in serial FS/LS or Carkit modes. NOT APPLICABLE
15	ULPIAUTORES	ULPI Auto Resume. Sets the AutoResume bit in Interface Control register on the ULPI PHY. • NO (0) = PHY does not use the AutoResume feature. • AUTO (1) = PHY uses the AutoResume feature.
14	Reserved	
13-10	USBTRDTIM	USB 2.0 Turnaround Time, in PHY clock cycles. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM).
9	Reserved	
8	ENBLSLPM	Enable UTMI Sleep. Controls assertion of utmi_sleep_n, utmi_l1_suspend_n outputs to the PHY when in the L1 state. • NO (0) = utmi_sleep_n, utmi_l1_suspend_n assertion from the core not transferred to the external PHY. • YES (1) = utmi_sleep_n, utmi_l1_suspend_n assertion from the core transferred to the external PHY.
7	PHYSEL	PHY Select. (HS vs. serial): Unused, since serial PHY is not supported.
6	SUSPHY	Suspend enable for USB2.0 HS/FS/LS PHY (ULPI or UTMI). Set to 1 only after core initialization is complete. • _0 (0) = USB2.0 PHY not placed in Suspend mode • _1 (1) = USB2.0 PHY enters Suspend mode if port is suspended or unused.
5	FSINTF	Full-Speed Serial Interface Select. UNUSED.
4	ULPI_UTMI_SEL	ULPI or UTMI+ Select • UTMI (0) = UTMI+ Interface • ULPI (1) = ULPI Interface
3	PHYIF	PHY Interface. DO NOT USE. If UTMI+ is selected, configures 8- or 16-bit interface. If ULPI is selected, configures SDR or DDR mode. • ZERO (0) = 8-bit UTMI @60 MHz / SDR (12-pin) ULPI • ONE (1) = 16-bit UTMI @30 MHz, not supported / DDR (8-pin) ULPI, not supported
2-0	TOUTCAL	HS/FS Timeout Calibration. The number of PHY clocks, as indicated by the application in this field, is multiplied by a bit-time factor; this factor is added to the high-speed/full-speed interpacket timeout duration in the core to account for additional delays introduced by the PHY. This may be required, since the delay introduced by the PHY in generating the linestate condition may vary among PHYs. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of connection. The number of bit times added per PHY clock are: High-speed operation: 8 bit times per 60-MHz PHY clock cycle Full-speed operation: (depending on clock speed) 0.2 bit times per 60-MHz PHY clock cycle 0.25 bit times per 48-MHz PHY clock cycle

3.2.8.28 GUSB2PHYACC [Offset = 0x1C280]

Global USB2 PHY Access

Figure 3-95. GUSB2PHYACC Register

31 27	26	25	24
Reserved	DISULPIDRVR	NEWREGREQ	VSTSDONE
R-0	R-0	R/W-0	R-0

23	22	21 16	15 14	13 8	7 0
VSTSBSY	REGWR	REGADDR	Reserved	EXTREGADDR	REGDATA
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-104. GUSB2PHYACC Register Field Descriptions

Bit	Name	Description
31-27	Reserved	
26	DISULPIDRVR	Disable ULPI Drivers, for carkit mode. Auto-cleared. NOT USED.
25	NEWREGREQ	New Register Request. Auto-cleared. • NEW (1) = Request new register access • NONE (0) = No action • ACTIVE (1) = Access request pending • IDLE (0) = No request pending
24	VSTSDONE	VStatus Done • DONE (1) = access is done • NOTDONE (0) = application has set the New Register Request bit, access is not done
23	VSTSBSY	VStatus Busy • DONE (0) = access is done • BUSY (1) = access is in progress
22	REGWR	Register Write • RD (0) = Read • WR (1) = Write
21-16	REGADDR	Register Address. ULPI PHY register address for immediate PHY Register Set access. Set to 6'h2F for Extended PHY Register Set access.
15-14	Reserved	
13-8	EXTREGADDR	ULPI: PHY extended register address. UTMI+: unused
7-0	REGDATA	Register Data (read and write data)

3.2.8.29 GUSB3PIPECTL [Offset = 0x1C2C0]

Global USB3 PIPE Control

Figure 3-96. GUSB3PIPECTL Register

31	30	28	27	26	25
PHYSOFTRST	Reserved		UX_EXIT_IN_PX	PING_ENHANCEMENT_EN	U1U2EXITFAIL_TO_RECOV
R/W-0	R/W-0		R/W-0	R/W-0	R/W-0

24	23	22	21	19
REQUEST_P1P2P3	STARTRXDETU3RXDET	DISRXDETU3RXDET	P1P2P3DELAY	
R/W-1	R/W-0	R/W-0	R/W-0x1	

18	17	16	15	14	13	12
DELAYP0TOP1P2P3	SUSPENDENABLE	DATWIDTH		ABORTRXDETINU2	SKIPRXDET	LFPSP0ALGN
R/W-0x1	R/W-0	R-0		R/W-0	R/W-0	R/W-0

11	10	9	8	7
P3P2TRANOK	P3EXSIGP2	LFPSFILT	Reserved	
R/W-0	R/W-0	R/W-0	R-0	

6	5	3	2	1	0
TXSWING	TXMARGIN		TXDEEMPHASIS		ELASTICBUFFERMODE
R/W-0	R/W-0		R/W-0		R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-105. GUSB3PIPECTL Register Field Descriptions

Bit	Name	Description
31	PHYSOFTRST	PHY Soft Reset. Active-high, fully static software reset for PIPE USB3 transceiver. • INACTIVE (0) = Reset inactive. • ACTIVE (1) = Reset active.
30-28	Reserved	
27	UX_EXIT_IN_PX	Workaround for SS PHY injecting a glitch on RxElecIdle while receiving Ux exit LFPS, and PowerDown change is in progress. • DEFAULT (0) = U1/U2/U3 exit done in PHY power state P0 • WA (1) = U1/U2/U3 exit done in PHY power state P1/P2/P3 respectively
26	PING_ENHANCEMENT_EN	Ping Enhancement Enable: Extended downstream port U1 ping receive timeout. Invalid for Upstream port. • DEFAULT (0) = U1 ping rx timeout is 300 ms (default) • _500 (1) = U1 ping rx timeout extended to 500 ms
25	U1U2EXITFAIL_TO_RECOV	Enhancement to prevent interoperability issue in case of incorrect LFPS handshake by the remote link. • DEFAULT (0) = LTSSM goes to SS Inactive when U1/U2 LFPS handshake fails (default) • ENHANCED (1) = LTSSM goes to Recovery when U1/U2 LFPS handshake fails, and to SS.Inactive only if recovery fails.
24	REQUEST_P1P2P3	Control the systematic request of P1/P2/P3 for U1/U2/U3 • NONE (0) = No P1/P2/P3 change request upon immediate Ux exit (remotely or locally initiated) • ALWAYS (1) = Always request transition from P0 to P1/P2/P3 on PHY during transition from U0 to U1/U2/U3 on USB.
23	STARTRXDETU3RXDET	Manual control for periodic Rx detection required in U3 and Rx.Detect, host mode. • NOOP (0) = No action • DETECT (1) = Trigger immediate Rx detection on transmitter.
22	DISRXDETU3RXDET	Disable the HW-scheduled periodic Rx detection required in U3 and SS.Disabled, for host mode. • AUTO (0) = Rx detection is HW-scheduled every 12/100 ms when in SS.Disabled / U3 LTSSM state, respectively. This requires the suspend clock to run permanently. • MANUAL (1) = HW-scheduled Rx detection is disabled, and must be SW-controlled using DoRxDetect.
21-19	P1P2P3DELAY	If DelayP0toP1P2P3=1, delays the transition to P1/P2/P3 when entering U1/U2/U3 until P1P2P3Delay*8b10b errors occur, or RxValid=0 on PIPE.

Table 3-105. GUSB3PIPECTL Register Field Descriptions (continued)

Bit	Name	Description
18	DELAYP0TOP1P2P3	Delay PHY change from P0 to P1/P2/P3 when link state changes from U0 to U1/U2/U3, respectively. • DIS (0) = When entering U1/U2/U3, transition to P1/P2/P3 without checking RxElecIlde, RxValid on PIPE. • EN (1) = When entering U1/U2/U3, delay the transition to P1/P2/P3 until RxElecIlde=1, RxValid=0 on PIPE
17	SUSPENDENABLE	Suspend Enable for USB3.0 SS PHY. Set to 1 only after core initialization is complete. • _0 (0) = USB3.0 PHY not placed in Suspend mode • _1 (1) = USB 3.0 PHY enters Suspend mode when conditions are valid.
16-15	DATWIDTH	PIPE Data Width (input from phy: refer to PIPE standard) Field updated to the input's value immediately after reset. • _32 (0) = PIPE data is 32 bit @125 MHz • _16 (1) = PIPE data is 16 bit @250 MHz • _8 (2) = PIPE data is 8 bit @500 MHz
14	ABORTRXDETINU2	Abort Rx Detect in U2. For Downstream port only. • NO_ABORT (0) = Receiver detection not aborted • ABORT (1) = When in U2, receiver detection will be aborted if U2 exit LFPS is received.
13	SKIPRXDET	Skip Rx Detect. When set, the core skips Rx Detection if pipe signal "RxElecIdle" is low. Skip is defined as waiting for the appropriate timeout, then repeating the operation.
12	LFPSP0ALGN	LFPS P0 Align. When set to 1: - The core deasserts LFPS transmission on the clock edge that it requests Phy power state 0 when exiting U1, U2, or U3 low power states. Otherwise, LFPS transmission is asserted one clock earlier. - The core requests symbol transmission two pipe_rx_pclks periods after the PHY asserts PhyStatus as a result of the PHY switching from P1 or P2 state to P0 state. • DEF (0) = Default • ALIGN (1) = DO NOT USE
11	P3P2TRANOK	P3-to-P2 Transitions OK • NOTSET (0) = P0 is always entered as an intermediate state during transitions between P2 and P3, as defined in the PIPE specification. • SET (1) = Core transitions directly from Phy power state P2 to P3 or from state P3 to P2. Illegal as per PIPE, use only if PHY cannot do LFPS in P3.
10	P3EXSIGP2	PHY power state behaviour upon U3 exit handshake. • DEFAULT (0) = Default behaviour • P2 (1) = Go to P2 before attempting a U3 exit handshake.
9	LFPSFILT	LFPS Filter. When set, filter LFPS reception with pipe "RxValid" signal in PHY power state P0, that is, ignore LFPS reception from the PHY unless both pipe signals "RxElecIdle" and "RxValid" are deasserted.
8-7	Reserved	
6	TXSWING	Tx Swing (output to phy: refer to PIPE standard)
5-3	TXMARGIN	Tx Margin[2:0] (output to phy: refer to PIPE standard)
2-1	TXDEEMPHASIS	Tx Deemphasis (output to phy: refer to PIPE standard) The value driven to the PHY is controlled by the LTSSM during USB3 Compliance mode.
0	ELASTICBUFFERMODE	Elastic Buffer Mode (output to phy: refer to PIPE standard)

3.2.8.30 GTXFIFOSIZ[N] [Offset = 0x1C300+4*N, N=0-15]

Global Transmit FIFO Size #N: FIFO mapping in RAM1, from staddr to (staddr+dep-1)

Figure 3-97. GTXFIFOSIZ Register

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-106. GTXFIFOSIZ Register Field Descriptions

Bit	Name	Description
31-16	TXFSTADDR	Transmit FIFO RAM Start Address, in 64-bit RAM words • MIN (0) = Minimum (RAM1 base) • MAX (1888) = Maximum (up to RAM1 depth - 32)
15-0	TXFDEP	Transmit FIFO Depth, in 64-bit RAM words • MIN (32) = Minimum depth • MAX (1920) = Maximum depth (the whole buffer)

3.2.8.31 GBL_RXFIFO_ARRAY [Offset = 50048]

Table 3-107. GBL_RXFIFO_ARRAY

Offset	Acronym	Register Description	Section
50048	GRXFIFOSIZ0	GRXFIFOSIZ0	Section 3.2.8.31.1
50052	GRXFIFOSIZ1	GRXFIFOSIZ1	Section 3.2.8.31.1
50056	GRXFIFOSIZ2	GRXFIFOSIZ2	Section 3.2.8.31.1

*3.2.8.31.1 GRXFIFOSIZ[N] [Offset = 0x1C380+4*N, N=0-2]*

Global Receive FIFO Size #N: FIFO mapping in RAM0, from staddr to (staddr+dep-1)

Figure 3-98. GRXFIFOSIZ Register

31	16	15	0
RXFSTADDR		RXFDEP	
R/W-0		R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-108. GRXFIFOSIZ Register Field Descriptions

Bit	Name	Description
31-16	RXFSTADDR	Receive FIFO RAM Start Address, in 64-bit RAM words. • MIN (0) = Minimum (RAM0 base) • MAX (962) = Maximum (base of descriptor cache in RAM0 - 32)
15-0	RXFDEP	Receive FIFO Depth, in 64-bit RAM words • MIN (32) = Minimum depth • MAX (994) = Maximum depth (the whole buffer)

3.2.8.32 GEVNTADR_LO[N] [Offset = 0x1C400+16*N, N=0-15]

Global Event Address: Lower 32 bits of start address of the external memory for the Event Buffer. During operation, hardware does not update this address.

Figure 3-99. GEVNTADR_LO Register

31	0
EVNTADR_LO	
R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-109. GEVNTADR_LO Register Field Descriptions

Bit	Name	Description
31-0	EVNTADR_LO	EVNTADR[31:0]

3.2.8.33 GEVNTADR_HI[N] [Offset = 0x1C404+16*N, N=0-15]

Global Event Address: Upper 32 bits of start address of the external memory for the Event Buffer. During operation, hardware does not update this address.

Figure 3-100. GEVNTADR_HI Register

31	0
EVNTADR_HI	
R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-110. GEVNTADR_HI Register Field Descriptions

Bit	Name	Description
31-0	EVNTADR_HI	EVNTADR[64:32]

3.2.8.34 GEVNTSIZ[N] [Offset = 0x1C408+16*N, N=0-15

Global Event Buffer Size

Figure 3-101. GEVNTSIZ Register

31	30 16	15 0
EVNTINTRPTMASK	Reserved	EVENTSIZ
R/W-0	R-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-111. GEVNTSIZ Register Field Descriptions

Bit	Name	Description
31	EVNTINTRPTMASK	Event Interrupt Mask. Prevents the interrupt from being generated when set to '1'. The events are queued wven when the mask is set.
30-16	Reserved	
15-0	EVENTSIZ	Event Buffer Size. Size of the Event Buffer, in bytes; must be a multiple of 4. Programmed by SW once during initialization.

3.2.8.35 GEVNTCOUNT[N] [Offset = 0x1C40C+16*N, N=0-15]

Global Event Buffer Count

Figure 3-102. GEVNTCOUNT Register

31 16	15 0
Reserved	EVNTCOUNT
R-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-112. GEVNTCOUNT Register Field Descriptions

Bit	Name	Description
31-16	Reserved	
15-0	EVNTCOUNT	Event Count. When read, returns the number of valid events in the Event Buffer (in bytes). When written, hardware decrements the count by the value written. The interrupt remains active while count is not 0.

3.2.8.36 GHWPARAMS8 [Offset = 0x1C600]

Global hardware parameters #8

Figure 3-103. GHWPARAMS8 Register

31	0
USB3_DCACHE_DEPTH_INFO	
R-0x000007F8	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-113. GHWPARAMS8 Register Field Descriptions

Bit	Name	Description
31-0	USB3_DCACHE_DEPTH_INFO	Depth of data cache, in 64-bit words (fixed). The cache occupies RAM0 from word 0 to DCACHE_DEPTH_INFO-1: Rx FIFOs shall be mapped from word DCACHE_DEPTH_INFO to RAM0_DEPTH-1.

3.2.9 Device Registers [Offset = 0x1C700]

Table 3-114. Device Registers

Offset	Acronym	Register Description	Section
0x1C700	DCFG	Device Configuration register	Section 3.2.9.1
0x1C704	DCTL	Device Control register	Section 3.2.9.2
0x1C708	DEVTEN	Device Event Enable register	Section 3.2.9.3
0x1C70C	DSTS	Device Status register	Section 3.2.9.4
0x1C710	DGCMDPAR	Device Generic Command Parameter register	Section 3.2.9.5
0x1C714	DGCMD	Device Generic Command	Section 3.2.9.6
0x1C720	DALEPENA	Device Active USB Endpoint Enable register	Section 3.2.9.7

3.2.9.1 DCFG [Offset = 0x1C700]

Device Configuration: configures the core in Device mode after power-on or after certain control commands or enumeration. Do not change after initial programming.

Figure 3-104. DCFG Register

31 24	23	22	21 17	16 12	11 10	9 3	2 0
Reserved	IGNORESTREAMPP	LPMCAP	NUMP	INTRNUM	PERFRINT	DEVADDR	DEVSPD
R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0x2	R/W-0	R/W-0x4

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-115. DCFG Register Field Descriptions

Bit	Name	Description
31-24	Reserved	
23	IGNORESTREAMPP	Ignore Packet-Pending for Stream management. From stream-capable bulk endpoints only. • NOCHANGE (0) = When Data Packet is received with PP=0 (OUT EP) or ACK is received with NumP=0 and PP=0 (IN EP), search another stream (CStream) to initiate to the host. This is non-optimal a) When host sets PP=0 even though it has not finished with the stream, and b) if the EP is configured with one transfer resource i.e. can't initiate another stream. • IGNORE (1) = PP is ignored for stream selection, no another stream is searched when DP is received with (PP=0) or ACK with (NumP=0, PP=0), to enhance performance if system bus bandwidth is low or host responds to ERDY quickly.

Table 3-115. DCFG Register Field Descriptions (continued)

Bit	Name	Description
22	LPMCAP	Link Power Management (LPM) Capability. • NO (0) = not LPM capable: device cannot respond to LPM transactions. • YES (1) = LPM capable: device shall respond to LPM transactions.
21-17	NUMP	Number of Receive Buffers. Indicates number of receive buffers to be reported in ACK TP. Value based on RxFIFO size, buffer sizes programmed in descriptors, and system latency.
16-12	INTRNUM	Interrupt Number. Interrupt/EventQ number on which non-endpoint-specific device related interrupts (see DEVT) are generated.
11-10	PERFRINT	Periodic Frame Interrupt. Time within a (micro)frame when the application must be notified using the End Of Periodic Frame Interrupt, which can be used to determine if all the periodic (isochronous, interrupt) traffic for that (micro)frame is complete. • _80 (0) = 80% of the (micro)frame interval • _85 (1) = 85% of the (micro)frame interval • _90 (2) = 90% of the (micro)frame interval • _95 (3) = 95% of the (micro)frame interval
9-3	DEVADDR	Device Address. Configure upon set-address USB command, clear to 0 upon USB reset. • DEF (0) = Default address for unenumerated device.
2-0	DEVSPD	Device Speed: USB speed at which the core should connect. Actual bus speed is determined only after chirp completion, based on the speed of the attached USB host. • HS (0) = High Speed (HS): 480 Mbit/s • FS (1) = Full Speed (FS): 12 Mbit/s • SS (4) = Super Speed (SS): 5 Gbit/s • LS_SERIAL (2) = Low Speed (LS): 1.5 Mbit/s on serial PHY: NOT SUPPORTED • FS_SERIAL (3) = Full Speed (FS): 12 Mbit/s on serial PHY: NOT SUPPORTED

3.2.9.2 DCTL [Offset = 0x1C704]

Device Control.

Figure 3-105. DCTL Register

31	30	29	28	27 24
RUNSTOP	CSFTRST	Reserved	HIRDTHRES_4	HIRDTHRES_TIME
R/W-0	R/W-0	R-0	R/W-0	R/W-0

23	22 20	19	18	17	16
APPL1RES	Reserved	KEEPCONNECT	L1HIBERNATIONEN	CRS	CSS
R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

15 13	12	11	10	9	8 5	4 1	0
Reserved	INITU2ENA	ACCEPTU2ENA	INITU1ENA	ACCEPTU1ENA	ULSTCHNGREQ	TSTCTL	Reserved
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-116. DCTL Register Field Descriptions

Bit	Name	Description
31	RUNSTOP	Run/Stop • STOP (0) = Stop USB device controller operation and signal disconnect to attached host. Clear only after all active transfers have been removed. For USB 3.0 disconnection, proceed by changing the target state to RX_DET. DSTS.DevCtrlHlt is set once disconect is complete and core is idle. • START (1) = Start USB device controller operation, after programming the device CSR. Controller will not signal connect to the attached host until the bit is set.

Table 3-116. DCTL Register Field Descriptions (continued)

Bit	Name	Description
30	CSFTRST	Core Soft Reset. Auto-cleared. The reset has the following effect: - Interrupts are cleared. - MMR are cleared except: GSTS, GCOREID, GGPIO, GUID, GUSB2PHYCFG, GUSB3PIPECTL, DCFG, DCTL, DEVTEN, DSTS. TxFIFOs and RxFIFO are flushed. - State machines are reset to the idle state, except the SoC slave. - Transactions on the SoC bus Master are terminated after completion. - Transactions on the USB are terminated immediately. • IDLE (0) = No ongoing reset, reset complete. • ACTIVE (1) = Reset ongoing • RESET (1) = Start reset (self-clearing)
29	Reserved	
28	HIRDTHRES_4	Host Initiated Resume Duration (HIRD) Threshold, MSbit: See HIRDTHRES_time
27-24	HIRDTHRES_TIME	Host Initiated Resume Duration (HIRD) Threshold, LSBits = timeout value. utmi_l1_suspend_n is asserted in L1 when : (HIRD value >= HIRDThres_time) and (HIRDThres_4=1) utmi_sleep_n is asserted in L1 when : (HIRD value < HIRDThres_time) or (HIRDThres_4=0) • _0 (0) = 60 us • _1 (1) = 135 us • _2 (2) = 210 us • _3 (3) = 285 us • _4 (4) = 360 us • _5 (5) = 435 us • _6 (6) = 510 us • _7 (7) = 585 us • _8 (8) = 660 us • _9 (9) = 735 us • _10 (10) = 810 us • _11 (11) = 885 us • _12 (12) = 960 us (maximum: higher values are invalid)
23	APPL1RES	LPM Response Programmed by Application: Handshake response made to LPM token. Note that if DCFG.LPMCap = 0, the response is always timeout (no response). • CAN_NYET (0) = NYET handshake to correct LPM transaction if data is pending in Tx FIFO or if OUT (Rx) endpoints are in flow controlled state. • ACK (1) = Core shall ACK correct LPM transaction regardless of Tx FIFO status and OUT (Rx) endpoint flow control.
22-20	Reserved	
19	KEEPCONNECT	Used for Save-and-Restore operation. DO NOT USE, SAR NOT IMPLEMENTED • NOACTION (0) = • KEEP (1) = Prevents device disconnection when DCTL.RunStop is cleared to 0. Enables the "Hibernation Request Event" when the link goes to U3/L2. Prevents LTSSM from automatically going to U0/L0 when host requests resume from U3/L2.
18	L1HIBERNATIONEN	DO NOT USE, SAR NOT IMPLEMENTED
17	CRS	Controller Restore State. DO NOT USE, SAR NOT IMPLEMENTED • RESTORE (0) = Initiate restore process, sets DSTS.RSS to 1 • NOACTION (1) =
16	CSS	Controller Save State. DO NOT USE, SAR NOT IMPLEMENTED • SAVE (0) = Initiate save process, sets DSTS.SSS to 1 • NOACTION (1) =
15-13	Reserved	

Table 3-116. DCTL Register Field Descriptions (continued)

Bit	Name	Description
12	INITU2ENA	Initiate U2 Enable. Cleared to 0 by USB reset. • NEWENUM1 (0) = May not initiate U2 • NEWENUM2 (1) = May initiate U2
11	ACCEPTU2ENA	Accept U2 Enable. Cleared to 0 by USB reset. • NO (0) = Reject U2 except when Force_LinkPM_Accept bit is set • YES (1) = Accept transition to U2 state if nothing is pending on the application side
10	INITU1ENA	Initiate U1 Enable. Cleared to 0 by USB reset. • NO (0) = May not initiate U1 • YES (1) = May initiate U1
9	ACCEPTU1ENA	Accept U1 Enable. Cleared to 0 by USB reset. • NO (0) = Reject U1 except when Force_LinkPM_Accept bit is set • YES (1) = Accept transition to U1 state if nothing is pending on the application side.
8-5	ULSTCHNGREQ	USB/Link State Change Request. A new request is indicated by a change of value. To issue the same request back-to-back, a 0 shall be written between the two requests. State change request result is reflected in DSTS. • NOOP (0) = No action • DIS (4) = SS: SS.Disabled • RXDET (5) = SS: Rx.Detect • INACT (6) = SS: SS.Inactive • REC (8) = SS: Recovery HS/FS/LS: Remote wakeup request. Issue only when device is in early suspend or suspend state • COMP (10) = SS: Compliance • LOOP (11) = SS: Loopback
4-1	TSTCTL	Test Control • _0 (0) = Test mode disabled • _1 (1) = Test_J mode • _2 (2) = Test_K mode • _3 (3) = Test_SE0_NAK mode • _4 (4) = Test_Packet mode • _5 (5) = Tes_Force_Enable mode
0	Reserved	

3.2.9.3 DEVTEN [Offset = 0x1C708]

Device Event Enable: enables the generation of Device-Specific events (see DEVT)

Figure 3-106. DEVTEN Register

31	14	13	12	11
Reserved		INACTTIMEOUTRCVEDEN	VNDRDEVTSTRCVEDEN	EVNTOVERFLOWEN
R-0		R/W-0	R/W-0	R/W-0

10	9	8	7	6
CMDCMPLTEN	ERRTICERREN	Reserved	SOFEN	EOPFEN
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

5	4	3	2	1	0
HIBERNATIONREQEVTEN	WKUPEVTEN	ULSTCNGEN	CONNECTDONEEN	USBRSTEN	DISCONNEVTEN
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-117. DEVTEN Register Field Descriptions

Bit	Name	Description
31-14	Reserved	
13	INACTTIMEOUTRCVEDEN	U2 Inactive Timeout Received Event Enable

Table 3-117. DEVTEN Register Field Descriptions (continued)

Bit	Name	Description
12	VNDRDEVTSTRCVEDEN	Vendor Device Test Received event Enable
11	EVNTOVERFLOWEN	Event Overflow event Enable
10	CMDCMPLTEN	Command Complete event Enable
9	ERRTICERREN	Erratic Error event Enable
8	Reserved	
7	SOFEN	Start of (micro)Frame event Enable. For debug only.
6	EOPFEN	End of Periodic Frame event Enable. For debug only.
5	HIBERNATIONREQEVTEN	Hibernation Request Event Enable. DO NOT USE, HIBERNATION NOT IMPLEMENTED
4	WKUPEVTEN	Resume/Remote Wakeup Detected Event Enable.
3	ULSTCNGEN	USB/Link State Change event Enable
2	CONNECTDONEEN	Connection Done event Enable
1	USBRSTEN	USB Reset Enable
0	DISCONNEVTEN	Disconnct Event Enable

3.2.9.4 DSTS [Offset = 0x1C70C]

Device Status

Figure 3-107. DSTS Register

31	30	29	28	27	26	25	24
Reserved		DCNRD	SRE	Reserved		RSS	SSS
R-0		R-0	R-0	R-0		R-0	R-0

23	22	21	18	17	16	3	2	0
COREIDLE	DEVCTRLHLT		USBLNKST	RXFIFOEMPTY	SOFFN		CONNECTSPD	
R-1	R-0		R-4	R-1	R-0		R-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-118. DSTS Register Field Descriptions

Bit	Name	Description
31-30	Reserved	
29	DCNRD	Device Controller Not Ready • WAIT (1) = Core is completing the state transitions after hibernation exit. Wait before processing DSTS.USBLnkSt • RDY (0) = Core is ready.
28	SRE	Save/Restore Error. NOT SUPPORTED.
27-26	Reserved	
25	RSS	Restore State Status, triggered by writing 1 to RSS • IDLE (0) = Restore is complete / inactive • RESTORING (1) = Restore is ongoing
24	SSS	Save State Status, triggered by writing 1 to CSS • IDLE (0) = Save is complete / inactive • SAVING (1) = Save is ongoing
23	COREIDLE	Core Idle status. asserted when all RxFIFO data transferred to system memory, all completed descriptors are written, and all Event Counts are zero. Changes after reset, so that reset value may not match first readout. • IDLE (1) = core is idle • ACTIVE (0) = core has unfinished activities

Table 3-118. DSTS Register Field Descriptions (continued)

Bit	Name	Description
22	DEVCTRLHLT	Device Controller Halted. Cleared (0) when the DCTL.RunStop is written to 1. Set (1) after DCTL.RunStop has been written to 0, core is idle and disconnect process is complete. When DevCtrlHlt =1, no Device events are generated.
21-18	USBLNKST	USB/Link State. Encoding depends on the connection speed (SS or HS/FS/LS) • _0 (0) = SS: LTSSM = U0 HS/FS/LS: On state • _1 (1) = SS: LTSSM = U1 • _2 (2) = SS: LTSSM = U2 HS/FS/LS: Sleep state • _3 (3) = SS: LTSSM = U3 HS/FS/LS: Suspend state • _4 (4) = SS: LTSSM = SS_DIS HS/FS/LS: Disconnected state (default) • _5 (5) = SS: LTSSM = RX_DET HS/FS/LS: Early Suspend state • _6 (6) = SS: LTSSM = SS_INACT • _7 (7) = SS: LTSSM = POLL • _8 (8) = SS: LTSSM = RECOV • _9 (9) = SS: LTSSM = HRESET • _10 (10) = SS: LTSSM = CMPLY • _11 (11) = SS: LTSSM = LPBK • _14 (14) = HS/FS/LS: Reset • _15 (15) = SS: LTSSM = Reset/Resume. USB resume or reset received from the host while in suspend. Write RECOV=8 to DCTL.ULStChngReq field to acknowledge it. HS/FS/LS: Resume
17	RXFIFOEMPTY	Rx FIFO Empty • NOTEMPTY (0) = Rx FIFO is not empty • EMPTY (1) = Rx FIFO is empty
16-3	SOFFN	received Start Of Frame's Frame Number
2-0	CONNECTSPD	Connection Speed. USB speed at which the device has come up after speed detection through a chirp sequence. • HS (0) = High Speed (HS): 480 Mbit/s • FS (1) = Full Speed (FS): 12 Mbit/s • SS (4) = Super Speed (SS): 5 Gbit/s • LS_SERIAL (2) = Low Speed (LS): 1.5 Mbit/s on serial PHY: NOT SUPPORTED • FS_SERIAL (3) = Full Speed (FS): 12 Mbit/s on serial PHY: NOT SUPPORTED

3.2.9.5 DGCMDPAR [Offset = 0x1C710]

Device Generic Command Parameter: To be programmed before or along with the device command itself.

Figure 3-108. DGCMDPAR Register

31	0
PARAMETER	
R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-119. DGCMDPAR Field Descriptions

Bit	Name	Description
31-0	PARAMETER	Parameter of the command; command-dependent.

3.2.9.6 DGCMD [Offset = 0x1C714]

Device Generic Command: generic command interface to send link management packets and notifications.

Figure 3-109. DGCMD Register

31 16	15	14 11	10	9	8	7 0
Reserved	CMDSTATUS	Reserved	CMDACT	Reserved	CMDIOC	CMDTYP
R-0	R-0	R-0	R/W-0	R-0	write-only-0	write-only-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-120. DGCMD Register Field Descriptions

Bit	Name	Description
31-16	Reserved	
15	CMDSTATUS	Command Status. • NONE (0) = No error • NEWENUM2 (1) = CmdErr: Error while processing the command.
14-11	Reserved	
10	CMDACT	Command active. Auto-cleared. • IDLE (0) = No ongoing command, command complete. • ACTIVE (1) = Command active (ongoing) • START (1) = Start generic command execution (self-clearing)
9	Reserved	
8	CMDIOC	Command Interrupt On Complete. Event mapped to interrupt # DCFG.IntNum (register). Reads return 0. • IOC (1) = Generic Command Completion event to be issued after executing the command.
7-0	CMDTYP	Command Type. Reads return 0. • _1 (1) = Transmit Set Link Function LPM. (Reserved in non-SuperSpeed operations) Parameter[0] = [Force_LinkPM_Accept] (1'b0: De-assert 1'b1: Assert) • _2 (2) = Set Periodic Parameters Parameter[9:0] (SystemExitLatency): Set value programmed by the host through the Set SEL device request, in microseconds. The Set SEL control transfer has 6 bytes of data and contains 4 values. Offset/Name/Meaning: 0 U1SEL Time in us for U1 System Exit Latency 1 U1PEL Time in us for U1 Device to Host Exit Latency 2 U2SEL Time in us for U2 System Exit Latency 4 U2PEL Time in us for U2 Device to Host Exit Latency If the device is enabled for U1 and U2, then the U2PEL should be programmed. If the device is enabled only for U1, then U1PEL should be programmed into this parameter. If the value is greater than 125 us, then the software must program a value of zero into this register. • _3 (3) = Transmit Function Wake Device Notification (Notification-only in SuperSpeed) Parameter[7:0] - Interface Number (IntfNum) that caused remote wakeup • _9 (9) = Selected FIFO Flush Parameter[4:0] = FIFO Number Paramer[5] = TX FIFO (1) or RX FIFO (0) • _10 (10) = All FIFO Flush. (No parameter) • _12 (12) = Set Endpoint NRDY: Makes the core think that the given endpoint is in an NRDY state. If buffers are available in that endpoint, an ERDY is immediately transmitted. Parameter[4:0] = Physical Endpoint Number • _16 (16) = Run SoC Bus LoopBack Test: Issue this command first, then the Start Transfer command to EP 0 and 1. Configure EP0 as OUT and EP1 as IN. The core reads data from the IN buffers and writes it back to OUT buffers. The IN and OUT must have an equal amount of data buffer. The endpoint 1 Tx-FIFO default value of (512/8)+2 = 66 should be changed to (1024/8)+2 = 130 for loopback mode. Parameter[0] = enable (1) and disable (0) Loopback mode. • _6 (6) = Transmit Function Host Role Request Device Notification (Notification-only in SuperSpeed) Parameter[1:0] = RSP Phase (Notification Type Specific) is INITIATE (2'b01) or CONFIRM (2'b10).

3.2.9.7 DALEPENA [Offset = 0x1C720]

Device Active USB Endpoint Enable. Set each bit (1) to enable the corresponding endpoint.

Bits 0 and 1 shall be set after USB reset, as they enable the control endpoint.

All other bits shall be set according to enumeration, and cleared upon a USB reset.

Figure 3-110. DALEPENA Register

31	30	29	28	27
USBACTEP15_IN	USBACTEP15_OUT	USBACTEP14_IN	USBACTEP14_OUT	USBACTEP13_IN
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
26	25	24	23	22
USBACTEP13_OUT	USBACTEP12_IN	USBACTEP12_OUT	USBACTEP11_IN	USBACTEP11_OUT
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
21	20	19	18	17
USBACTEP10_IN	USBACTEP10_OUT	USBACTEP9_IN	USBACTEP9_OUT	USBACTEP8_IN
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
16	15	14	13	12
USBACTEP8_OUT	USBACTEP7_IN	USBACTEP7_OUT	USBACTEP6_IN	USBACTEP6_OUT
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
11	10	9	8	
USBACTEP5_IN	USBACTEP5_OUT	USBACTEP4_IN	USBACTEP4_OUT	
R/W-0	R/W-0	R/W-0	R/W-0	
7	6	5	4	
USBACTEP3_IN	USBACTEP3_OUT	USBACTEP2_IN	USBACTEP2_OUT	
R/W-0	R/W-0	R/W-0	R/W-0	
3	2	1	0	
USBACTEP1_IN	USBACTEP1_OUT	USBACTEP0_IN	USBACTEP0_OUT	
R/W-0	R/W-0	R/W-0	R/W-0	

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-121. DALEPENA Register Field Descriptions

Bit	Name	Description
31	USBACTEP15_IN	USB Activate Endpoint 15 IN
30	USBACTEP15_OUT	USB Activate Endpoint 15 OUT
29	USBACTEP14_IN	USB Activate Endpoint 14 IN
28	USBACTEP14_OUT	USB Activate Endpoint 14 OUT
27	USBACTEP13_IN	USB Activate Endpoint 13 IN
26	USBACTEP13_OUT	USB Activate Endpoint 13 OUT
25	USBACTEP12_IN	USB Activate Endpoint 12 IN
24	USBACTEP12_OUT	USB Activate Endpoint 12 OUT
23	USBACTEP11_IN	USB Activate Endpoint 11 IN
22	USBACTEP11_OUT	USB Activate Endpoint 11 OUT
21	USBACTEP10_IN	USB Activate Endpoint 10 IN
20	USBACTEP10_OUT	USB Activate Endpoint 10 OUT
19	USBACTEP9_IN	USB Activate Endpoint 9 IN
18	USBACTEP9_OUT	USB Activate Endpoint 9 OUT
17	USBACTEP8_IN	USB Activate Endpoint 8 IN
16	USBACTEP8_OUT	USB Activate Endpoint 8 OUT
15	USBACTEP7_IN	USB Activate Endpoint 7 IN
14	USBACTEP7_OUT	USB Activate Endpoint 7 OUT
13	USBACTEP6_IN	USB Activate Endpoint 6 IN
12	USBACTEP6_OUT	USB Activate Endpoint 6 OUT
11	USBACTEP5_IN	USB Activate Endpoint 5 IN
10	USBACTEP5_OUT	USB Activate Endpoint 5 OUT
9	USBACTEP4_IN	USB Activate Endpoint 4 IN
8	USBACTEP4_OUT	USB Activate Endpoint 4 OUT
7	USBACTEP3_IN	USB Activate Endpoint 3 IN

Table 3-121. DALEPENA Register Field Descriptions (continued)

Bit	Name	Description
6	USBACTEP3_OUT	USB Activate Endpoint 3 OUT
5	USBACTEP2_IN	USB Activate Endpoint 2 IN
4	USBACTEP2_OUT	USB Activate Endpoint 2 OUT
3	USBACTEP1_IN	USB Activate Endpoint 1 IN
2	USBACTEP1_OUT	USB Activate Endpoint 1 OUT
1	USBACTEP0_IN	USB Activate Endpoint 0 IN (Control)
0	USBACTEP0_OUT	USB Activate Endpoint 0 OUT (Control)

3.2.10 USB2.0 OTG and Battery Charger Registers [Offset = 0x1CC00]

Table 3-122. USB2.0 OTG Registers

Offset	Acronym	Register Description	Section
0x1CC00	OCFG	OTG Configuration register	Section 3.2.10.1
0x1CC04	OCTL	OTG Control register	Section 3.2.10.2
0x1CC08	OEVT	OTG Event register	Section 3.2.10.1
0x1CC0C	OEVTEN	OTG Event Enable register	Section 3.2.10.4
0x1CC10	OSTS	OTG Status register	Section 3.2.10.5
0x1CC20	ADPCFG	ADP Configuration register	Section 3.2.10.6
0x1CC24	ADPCTL	ADP Control register	Section 3.2.10.7
0x1CC28	ADPEVT	ADP Event register	Section 3.2.10.8
0x1CC2C	ADPEVTEN	ADP Event Enable register	Section 3.2.10.9

3.2.10.1 OCFG [Offset = 0x1CC00]

OTG configuration.

Figure 3-111. OCFG Register

31	4	3	2	1	0
Reserved		OTGSFTRSTMSK	OTGVERSION	HNPCAP	SRPCAP
R-0		R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-123. OCFG Register Field Descriptions

Bit	Name	Description
31-4	Reserved	
3	OTGSFTRSTMSK	Protects OTG, PHY and VBUS filters from the following SW resets: xHCI USBCMD.HCRST (host), DCTL.CSftRst (device). Note: In OTG2 applications, it is not recommended to program USBCMD.HCRST during role switch. • DEFAULT (0) = OTG logic reset by SW resets • MASK (1) = OTG logic not affected by SW resets. Use only with GCTL.PrtCapDir = 2'b11
2	OTGVERSION	Debug, always write 0.
1	HNPCAP	HNP Capabilty Enable. • NO (0) = HNP capability is disabled • YES (1) = HNP capability is enabled
0	SRPCAP	SRP Capability enable. For A-device, SRP detection. For B-device, SRP generation. • NO (0) = SRP capability is disabled • YES (1) = SRP capability is enabled

3.2.10.2 OCTL [Offset = 0x1CC04]

OTG control.

IMPORTANT NOTE: register is reinitialized upon ID change, but is not affected by a software reset.

Figure 3-112. OCTL Register

31	8	7	6	5
Reserved		OTG3_GOERR	PERIMODE	PRTPWRCTL
R-0		R/W-0	R/W-1	R/W-0

4	3	2	1	0
HNPREQ	SESREQ	TERMSELDLPULSE	DEVSETHNPEN	HSTSETHNPEN
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-124. OCTL Register Field Descriptions

Bit	Name	Description
31-8	Reserved	
7	OTG3_GOERR	To be set upon TRSP_ACK_ERR, TRSP_CNF_ERR, or TRSP_WRST_ERR timeout. Auto-cleared. OTG3: NOT IMPLEMENTED, DO NOT SET. • GO (1) = Sends LTSSM to error state during OTG 3.0 RSP • PENDING (1) = LTSSM transition to error state is pending • NOOP (0) =
6	PERIMODE	Peripheral Mode. Program the core to work as a peripheral or as a host. • NO (0) = OTG device acts as a host • YES (1) = OTG device acts as a peripheral
5	PRTPWRCTL	Port Power Control. Set or cleared by SW. Self-cleared in any of the following conditions: 1) transition to a_idle OTG state 2) aidl_bdis_tout event when in a_suspend OTG state 3) a_wait_bcon_tout event when in a_wait_bcon OTG state 4) transition to any b_* OTG state • REQ (1) = Initiate the VBUS drive on the USB, when A-device. • ON (1) = VBUS is driven. • OFF (0) = VBUS drive is off • SWOFF (0) = Manually switch off VBUS drive.
4	HNPREQ	HNP Request. Set (1) by SW to initiate HNP request to the connected USB host. Clear (0) by SW upon either OEVT.OTGBDevBHostEndEvnt or OEVT.OTGBDevVBusChngEvnt. • ONGOING (1) = HNP request active. • DONE (0) = HNP request inactive.
3	SESREQ	Session Request. In the absence of OEVT.OTGBDevSessVldDetEvnt after a request, the application must wait for at least TB_SRP_FAIL (6 secs) before another request. • SRP (1) = Initiate the SRP (data line pulsing) on the USB. • NOOP (0) = No action • ZERO (0) = All reads return zero.
2	TERMSELDLPULSE	TermSelect Data Line Pulse. Alternate SRP data line pulsing method on UTMI interface. • NEWENUM1 (0) = utmi_txvalid used for SRP generation (default) • NEWENUM2 (1) = utmi_termselect used for SRP generation
1	DEVSETHNPEN	Device Set HNP Enable. To be set when HNP has been successfully enabled by the connected host, using the SetFeature.SetHNPEnable command. • DIS (0) = HNP disabled in device • EN (1) = HNP enabled in device
0	HSTSETHNPEN	Host Set HNP Enable. To be set when HNP has been successfully enabled on the connected device, using the SetFeature.SetHNPEnable command. • DIS (0) = HNP disabled in host • EN (1) = HNP enabled in host

3.2.10.3 OEVT [Offset = 0x1CC08

OTG Event: OTG interrupt status. All writable bits are cleared by writing a 1.

Figure 3-113. OEVT Register

31	30	25	24	23
DEVICEMODE	Reserved		OTGCONIDSTSCHNGEVNT	HRRCONFNOTIFEVNT
R-0	R-0		R/W-0	R/W-0

22	21	20	19
HRRINITNOTIFEVNT	OTGADEVIDLEEVNT	OTGADEVBHOSTENDEVNT	OTGADEVHOSTEVNT
R/W-0	R/W-0	R/W-0	R/W-0

18	17	16
OTGADEVHNPCHNGDETEVNT	OTGADEVSRPDETEVNT	OTGADEVSESSENDDETEVNT
R/W-0	R/W-0	R/W-0

15	12	11	10	9
Reserved		OTGBDEVHOSTENDEVNT	OTGBDEVHNPCHNGEVNT	OTGBDEVSESSVLDDETEVNT
R-0		R/W-0	R/W-0	R/W-0

8	7	4	3	2	1	0
OTGBDEVVBUSCHNGEVNT	Reserved		BSESVLD	HSTNEGSTS	SESREQSTS	OEVTERROR
R/W-0	R-0		R-0	R-0	R-0	R/W-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-125. OEVT Register Field Descriptions

Bit	Name	Description
31	DEVICEMODE	Dual-role device's mode, based on iddig input. • A (0) = A-device = default host • B (1) = B-device = default peripheral
30-25	Reserved	
24	OTGCONIDSTSCHNGEVNT	Connector ID status change event. Set in both A-device and B-device mode. • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action
23	HRRCONFNOTIFEVNT	Host Role Request Confirm Notifier Event. Set upon reception of HRR Device Notification TP with Confirm field set. Set in OTG3, SS, A-host or B-host mode only. OTG3: NOT IMPLEMENTED • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action
22	HRRINITNOTIFEVNT	Host Role Request Initiate Notifier Event. Set upon reception of HRR Device Notification TP with Initiate field set. Set in OTG3, SS, A-host or B-host mode only. OTG3: NOT IMPLEMENTED • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action

Table 3-125. OEVT Register Field Descriptions (continued)

Bit	Name	Description
21	OTGADEVIDLEEVNT	A-device A-IDLE Event. Set when OTG FSM enters A-IDLE state from any other state. Set in A-device mode only. OTG3: NOT IMPLEMENTED • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action
20	OTGADEVBHOSTENDEVNT	A-device B-host End Event. Set when connected B-device has completed its B-host role and returns to B-peripheral. Set in A-device mode only. • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action
19	OTGADEVHOSTEVNT	A-device Host Event. Set when device enters host role, upon initial connect to B-device as well as upon HNP from A-peripheral to A-host. Set in A-device mode only. • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action
18	OTGADEVHNPCHNGDETEVNT	A-device HNP change Detected Event. Set when there is an HNP event. Set in A-device mode only. • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action
17	OTGADEVSRPDETEVNT	A-device SRP Detected Event. Set when SRP request from B-device is detected. Set in A-device mode only. • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action
16	OTGADEVSESSENDDETEVNT	A-device Session End Detected Event. Set when UTMI input "a-vbus-valid" is deasserted (0). Set in A-device mode only. • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action
15-12	Reserved	
11	OTGBDEVHOSTENDEVNT	B-device Host End Event. Set completing B-host role and returning to default B-peripheral role. Set in B-device mode only. • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action
10	OTGBDEVHNPCHNGEVNT	B-device HNP Change Event. Set upon (success of failure of an) HNP attempt. Set in B-device mode only. • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action

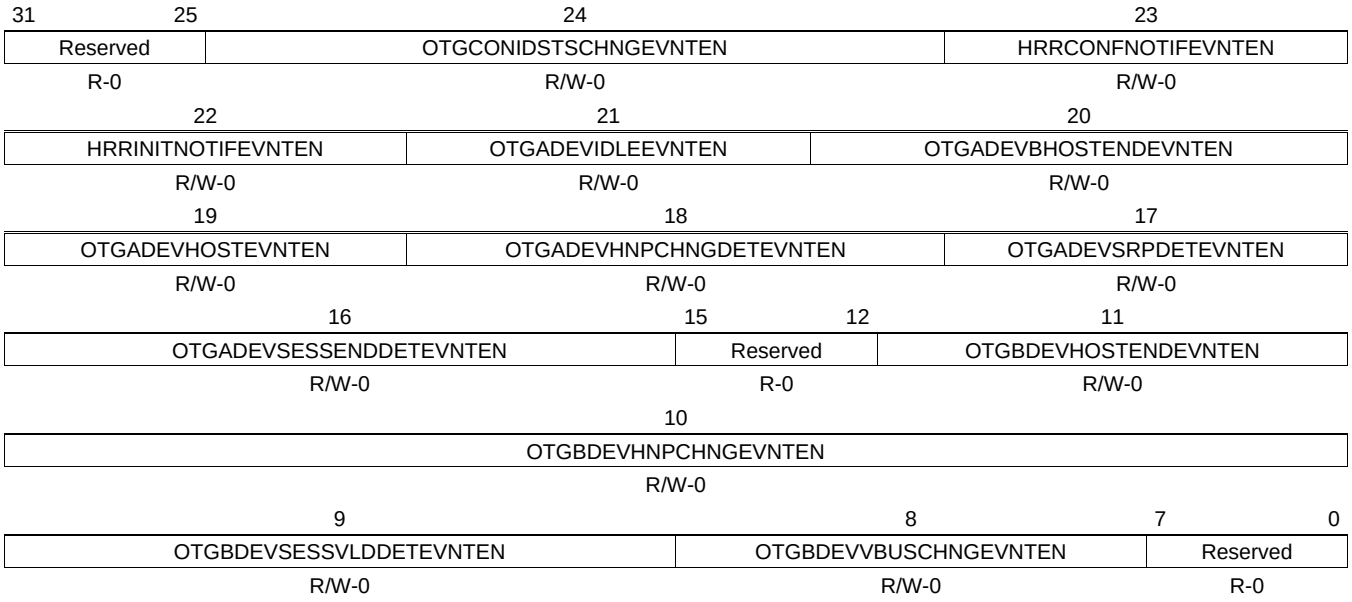
Table 3-125. OEVT Register Field Descriptions (continued)

Bit	Name	Description
9	OTGBDEVSESSVLDDETEVNT	B-device Session Valid Detected Event. Set when B-device succeeds in starting a session. Set in B-device mode only. • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action
8	OTGBDEVVBUSCHNGEVNT	B-device VBUS Change Event. Set when UTMI input "b-session-valid" transitions (to 0 or 1). Set in B-device mode only. • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action
7-4	Reserved	
3	BSESVLD	B-Session Valid. Updated when OTGBDevVBUSChngEvnt is set. • INVALID (0) = B-session is not valid on VBUS • VALID (1) = B-session is valid on VBUS
2	HSTNEGSTS	Host Negociation Status. Updated when OTGADevHNPChngEvnt or OTGBDevHNPChngEvnt is set. • FAILURE (0) = Host negotiation failure. In A-device, indicates imminent end of session indication from core. In B-device, it indicates that the timer used to wait for an A-device to signal a connection (b_ase0_brst_tmout) timed out resulting in B-device staying as B-peripheral. • SUCCESS (1) = Host negotiation success.
1	SESREQSTS	Session Request Status. Updated when OTGBDevSessVldDetEvnt is set. • NOSRP (0) = Session not started as a result of SRP. • SRP (1) = Session started as a result of successful SRP.
0	OEVTERROR	No errors currently defined. • NOEVT (0) = Event was not set • EVT (1) = Event was set by core • CLR (1) = Clear the event • NOOP (0) = No action

3.2.10.4 OEVTEN [Offset = 0x1CC0C]

OTG Event Enable: OTG interrupt event enable.

Figure 3-114. OEVTEN Register



Legend: R = Read only; W = Write only; -n = value after reset

Table 3-126. OEVTEN Register Field Descriptions

Bit	Name	Description
31-25	Reserved	
24	OTGCONIDSTSCHNGEVNTEN	Connector ID Status Change Event Enable. • DIS (0) = Event disabled • EN (1) = Event enabled
23	HRRCONFNOTIFEVNTEN	Host Role Request Confirm Notifier Event Enable. OTG3: NOT IMPLEMENTED • DIS (0) = Event disabled • EN (1) = Event enabled
22	HRRINITNOTIFEVNTEN	Host Role Request Initiate Notifier Event Enable. OTG3: NOT IMPLEMENTED • DIS (0) = Event disabled • EN (1) = Event enabled
21	OTGADEVIDLEEVNTEN	A-device A-IDLE Event Enable. • DIS (0) = Event disabled • EN (1) = Event enabled
20	OTGADEVBHOSTENDEVNTEN	A-device B-host End Event Enable. • DIS (0) = Event disabled • EN (1) = Event enabled
19	OTGADEVHOSTEVNTEN	A-device Host Event Enable. • DIS (0) = Event disabled • EN (1) = Event enabled
18	OTGADEVHNPCHNGDETEVNTEN	A-device HNP change Detected Event Enable. • DIS (0) = Event disabled • EN (1) = Event enabled
17	OTGADEVSRPDETEVNTEN	A-device SRP Detected Event Enable. • DIS (0) = Event disabled • EN (1) = Event enabled

Table 3-126. OEVTEN Register Field Descriptions (continued)

Bit	Name	Description
16	OTGADEVSESSENDDETEVNTEN	A-device Session End Detected Event Enable. • DIS (0) = Event disabled • EN (1) = Event enabled
15-12	Reserved	
11	OTGBDEVHOSTENDEVNTEN	B-device Host End Event Enable. • DIS (0) = Event disabled • EN (1) = Event enabled
10	OTGBDEVHNPCHNGEVNTEN	B-device HNP Change Event Enable. • DIS (0) = Event disabled • EN (1) = Event enabled
9	OTGBDEVSESSVLDDETEVNTEN	B-device Session Valid Detected Event Enable. • DIS (0) = Event disabled • EN (1) = Event enabled
8	OTGBDEVVBUSCHNGEVNTEN	B-device VBUS Change Event Enable. • DIS (0) = Event disabled • EN (1) = Event enabled
7-0	Reserved	

3.2.10.5 OSTS [Offset = 0x1CC10]

OTG status.

Figure 3-115. OSTS Register

31 12	11 8	7 5	4	3	2	1	0
Reserved	OTGSTATE	Reserved	PERIPHERALSTATE	XHCIPRTPOWER	BSESVLD	VBUSVLD	CONIDSTS
R-0	R-0x8	R-0	R-1	R-1	R-0	R-0	R-1

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-127. OSTS Register Field Descriptions

Bit	Name	Description
31-12	Reserved	
11-8	OTGSTATE	OTG state machine state, for debug. Default value may vary depending on integration. • _0 (0) = a_idle • _1 (1) = a_wait_vrise • _2 (2) = a_wait_bcon • _3 (3) = a_wait_vfall • _4 (4) = a_vbus_err • _5 (5) = a_host • _6 (6) = a_suspend • _15 (7) = a_peripheral • _7 (8) = a_wait_ppwr • _8 (9) = b_idle • _9 (10) = b_srp_init • _10 (11) = b_peripheral • _11 (12) = b_wait_acon • _12 (13) = b_host • _13 (14) = a_wait_switch • _14 (15) = b_wait_switch
7-5	Reserved	

Table 3-127. OSTS Register Field Descriptions (continued)

Bit	Name	Description
4	PERIPHERALSTATE	Current role of the controller • HOST (0) = Host • PERIPH (1) = Peripheral
3	XHCIPRTPOWER	xHCI host Port Power. Reflects host bitfiled PORTSC.PP
2	BSESVLD	VBUS B-Session Valid status • NO (0) = B-session is not valid • YES (1) = B-session is valid
1	VBUSVLD	VBUS Valid status • NO (0) = VBUS is not valid • YES (1) = VBUS is valid
0	CONIDSTS	Connector ID Status. Default value may vary depending on integration. • A (0) = Core is A-device • B (1) = Core is B-device

3.2.10.6 ADPCFG [Offset = 0x1CC20]

ADP Configuration Register

Figure 3-116. ADPCFG Register

31 30	29 28	27 26	25 0
PRBPER	PRBDELTA	PRBDSCHG	Reserved
R/W-0	R/W-0	R/W-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-128. ADPCFG Register Field Descriptions

Bit	Name	Description
31-30	PRBPER	These bits sets the T_ADP_PRB as follows: 2'b00 -0.775 sec 2'b01 - 1.55 sec 2'b10 - .275 sec 2'b11 - Reserved scaledownThe scaledown values of PrbPer are prb_per== 2'b00 => 12.50 ms prb_per== 2'b01 => 18.75 ms prb_per== 2'b10 => 25.00 ms prb_per== 2'b11 => 31.25 ms
29-28	PRBDELTA	These bits set the resolution for RTIM value. The bits are defined in units of 32 kHz clock cycles as follows: 2'b00 - 1 cycles 2'b01 - 2 cycles 2'b10 - 3 cycles 2'b11 - 4 cycles For example if this value is chosen to 2'b01, it means that RTIM increments for every two 32Khz clock cycles.

Table 3-128. ADPCFG Register Field Descriptions (continued)

Bit	Name	Description
27-26	PRBDSCHG	These bits set the times for TadpDschg. These bits are defined as follows: 2'b00- 4 msec 2'b01- 8 msec 2'b10- 16 msec 2'b11- 32 msec The scaledown values for the PrbDschg are 2'b00: 62.5 us 2'b01: 125 us 2'b10: 250 us 2'b11: 500 us
25-0	Reserved	

3.2.10.7 ADPCTL [Offset = 0x1CC24]

ADP Control Register

Figure 3-117. ADPCTL Register

31 29	28	27	26	25	24	23 0
Reserved	ENAPRB	ENASNS	ADPEN	ADPRES	WB	Reserved
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-129. ADPCTL Register Field Descriptions

Bit	Name	Description
31-29	Reserved	
28	ENAPRB	When programmed to 1'b1 along with ADPEn, the core performs a probe operation.
27	ENASNS	When programmed to 1'b1 along with ADPEn, the core performs a sense operation.
26	ADPEN	When set, the core performs either ADP probing or sensing based on EnaPrb or EnaSns. ADPEn = 0 gates the ADP clock for major portion of ADP related logic
25	ADPRES	When set, the core performs either ADP probing or sensing based on EnaPrb or EnaSns. ADPEn = 0 gates the ADP clock for major portion of ADP related logic
24	WB	Write Busy 1'b0 Write Completed 1'b1 Write in Progress The application can read or write ADPCFG and ADPCTL registers only if this field is cleared. The HW sets this bit when the write is in progress in Suspend clock domain.
23-0	Reserved	

3.2.10.8 ADPEVT [Offset = 0x1CC28]

ADP Event Register

Writing 1 to the info bit in this register clears the register bit and associated ADP interrupt

Figure 3-118. ADPEVT Register

31 29	28	27	26	25	24 11	10 0
Reserved	ADPPRBEVNT	ADPSNSEVNT	ADPTMOUTEVNT	ADPRSTCMPLTEVNT	Reserved	RTIM
R-0	R-0	R-0	R-0	R-0	R-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-130. ADPEVT Register Field Descriptions

Bit	Name	Description
31-29	Reserved	
28	ADPPRBEVNT	When this Event is set, it means that the VBUS voltage is greater than VadpPrb or VadpPrb is reached.
27	ADPSNSEVNT	When this Event is set, it means that the VBUS voltage is greater than VadpSns value or VadpSns is reached.
26	ADPTMOUTEVNT	This Event is relevant when ADP probe command is executed. When this Event is set, it means that the ramp time has completed (GADPCTL.RTIM has reached its terminal value of 0x7FF). This is a debug feature that allows software to read the ramp time after each cycle.
25	ADPRSTCMPLTEVNT	This event, when set, indicates that the ADP Reset command is successful
24-11	Reserved	
10-0	RTIM	These bits capture the latest time it took for VBUS to ramp from VADP_SINK to VADP_PRB. The bits are defined in units of 32 kHz clock cycles as follows: 0x000 - 1 cycles 0x001 - 2 cycles 0x002 - 3 cycles and so on till 0x7FF - 2048 cycles A time of 1024 cycles at 32 kHz corresponds to a time of 32 msec. (Note for scaledown ramp_timeout = PrbDelta == 2'b00 => 6250 us PrbDelta == 2'b01 => 3125 us PrbDelta == 2'b01 => 1562.5 us PrbDelta == 2'b01 => 781.25 us

3.2.10.9 ADPEVTEN [Offset = 0x1CC2C]

ADP Event Enable Register

Figure 3-119. ADPEVTEN Register

31 29	28	27	26	25	24 0
Reserved	ADPPRBEVNTEN	ADPSNSEVNTEN	ADPTMOUTEVNTEN	ADPRSTCMPLTEVNTEN	Reserved
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

Legend: R = Read only; W = Write only; -n = value after reset

Table 3-131. ADPEVTEN Register Field Descriptions

Bit	Name	Description
31-29	Reserved	
28	ADPPRBEVNTEN	When this bit is set, ADPPrbEvnt in ADPEVT register is enabled.
27	ADPSNSEVNTEN	When this bit is set, AdpSnsEvntEn in ADPEVT register is enabled.
26	ADPTMOUTEVNTEN	When this bit is set, AdpTmoutEvntEn in ADPEVT register is enabled.
25	ADPRSTCMPLTEVNTEN	When this bit is set, ADPRstCmpltEvnt in ADPEVT register is enabled.
24-0	Reserved	

3.3 USB PHY Control Registers

NOTE: The following registers reside in the Bootcfg module of K2 devices. Their base address is device dependent and user needs to refer to the specific device's data sheet for it. As an example, on K2K device, it is 0x02620738.

Table 3-132. PHY Control Registers (host)

Offset	Acronym	Register Description	Section
0x0000	USB_PHY_CTL0 Register	USB PHY Control register 0	Section 3.3.1
0x0004	USB_PHY_CTL1 Register	USB PHY Control register 1	Section 3.3.2
0x0008	USB_PHY_CTL2 Register	USB PHY Control register 2	Section 3.3.3
0x000C	USB_PHY_CTL3 Register	USB PHY Control register 3	Section 3.3.4
0x0010	USB_PHY_CTL4 Register	USB PHY Control register 4	Section 3.3.5
0x0014	USB_PHY_CTL5 Register	USB PHY Control register 5	Section 3.3.6

3.3.1 USB_PHY_CTL0 Register [Offset = 0x0000]

Figure 3-120. USB_PHY_CTL0 Register

31					12	11
Reserved						PHY_RTUNE_ACK
R-0						R-0
10	9	8		7	6	5
PHY_RTUNE_REQ	Reserved	PHY_TC_VATESTENB			PHY_TC_TEST_POWERDOWN_SSP	PHY_TC_TEST_POWERDOWN_HSP
R/W-0	R-0	R/W-00			R/W-0	R/W-0
4	3	2	1	0		
PHY_TC_LOOPBACKENB	Reserved	UTMI_VBAUSVLDEXT	UTMI_TXBITSTUFFENH	UTMI_TXBITSTUFFEN		
R/W-0	R-0	R/W-0	R/W-0	R/W-0		

Legend: R = Read only; W = Write only; - *n* = value after reset

Table 3-133. USB_PHY_CTL0 Register Field Descriptions

Bit	Field	Description
31-12	Reserved	Reserved
11	PHY_RTUNE_ACK	The PHY uses an external resistor to calibrate the termination impedances of the PHY's high-speed inputs and outputs. The resistor is shared between the USB2.0 high-speed outputs and the Super-speed I/O. Each time the PHY is taken out of a reset, a termination calibration is performed. For SS link, the calibration can also be requested externally by asserting the PHY_RTUNE_REQ When the calibration is complete, the PHY_RTUNE_ACK transitions low. A resistor calibration on the SS link cannot be performed while the link is operational
10	PHY_RTUNE_REQ	See PHY_RTUNE_ACK.
9	Reserved	Reserved
8-7	PHY_TC_VATESTENB	Analog Test Pin Select. Enables analog test voltages to be placed on the ID pin. • 11 = Invalid setting. • 10 = Invalid setting. • 01 = Analog test voltages can be viewed or applied on ID. • 00 = Analog test voltages cannot be viewed or applied on ID.
6	PHY_TC_TEST_POWERDOWN_SSP	SS Function Circuits Power-Down Control. Powers down all SS function circuitry in the PHY for IDDQ testing.
5	PHY_TC_TEST_POWERDOWN_HSP	HS Function Circuits Power-Down Control Powers down all HS function circuitry in the PHY for IDDQ testing.

Table 3-133. USB_PHY_CTL0 Register Field Descriptions (continued)

Bit	Field	Description
4	PHY_TC_LOOPBACKENB	Loop-back Test Enable Places the USB3.0 PHY in HS Loop-back mode, which concurrently enables the HS receive and transmit logic. • 1 = During HS data transmission, the HS receive logic is enabled. • 0 = During HS data transmission, the HS receive logic is disabled.
3	Reserved	• Reserved
2	UTMI_VBAUSVLDEXT	External VBUS Valid Indicator Function: Valid in Device mode and only when the VBUSVLDEXTSEL signal is set to 1'b1. VBUSVLDEXT indicates whether the VBUS signal on the USB cable is valid. In addition, VBUSVLDEXT enables the pull-up resistor on the D+ line. • 1 = VBUS signal is valid, and the pull-up resistor on D+ is enabled. • 0 = VBUS signal is not valid, and the pull-up resistor on D+ is disabled.
1	UTMI_TXBITSTUFFENH	High-byte Transmit Bit-Stuffing Enable Function: controls bit stuffing on DATAINH[7:0] when OPMODE[1:0]=11b. • 1 = Bit stuffing is enabled. • 0 = Bit stuffing is disabled.
0	UTMI_TXBITSTUFFEN	Low-byte Transmit Bit-Stuffing Enable Function: controls bit stuffing on DATAIN[7:0] when OPMODE[1:0]=11b. • 1 = Bit stuffing is enabled. • 0 = Bit stuffing is disabled.

3.3.2 USB_PHY_CTL1 Register [Offset = 0x0004]

Figure 3-121. USB_PHY_CTL1 Register

31	6	5
Reserved		PIPE_REF_CLKREQ_N
R-0		R-0

4	3	2	1	0
PIPE_TX2RX_LOOPBK	PIPE_EXT_PCLK_REQ	PIPE_ALT_CLK_SEL	PIPE_ALT_CLK_REQ	PIPE_ALT_CLK_EN
R/W-0	R/W-0	R/W-0	R-0	R/W-0

Legend: R = Read only; R/W = Read/Write, - *n* = value after reset

Table 3-134. USB_PHY_CTL1 Register Field Descriptions

Bit	Field	Description
31-6	Reserved	Reserved
5	PIPE_REF_CLKREQ_N	Reference Clock Removal Acknowledge. When the pipeP_power-down control into the PHY turns off the MPLL in the P3 state, PIPE_REF_CLKREQ_N is asserted after the PLL is stable and the reference clock can be removed.
4	PIPE_TX2RX_LOOPBK	Loop-back. When this signal is asserted, data from the transmit predriver is looped back to the receiver slicers. LOS is bypassed and based on the tx_en input so that rx_los=!tx_data_en.
3	PIPE_EXT_PCLK_REQ	External PIPE Clock Enable Request. When asserted, this signal enables the pipeP_pclk output regardless of power state (along with the associated increase in power consumption).
2	PIPE_ALT_CLK_SEL	Alternate CLock Source Select. Selects the alternate clock sources instead of the internal MPLL outputs for the PCS clocks. • 1 = Uses alternate clocks. • 0 = Users internal MPLL clocks. Change only during a reset. Note: It is reserved bit on K2E and K2L devices.

Table 3-134. USB_PHY_CTL1 Register Field Descriptions (continued)

Bit	Field	Description
1	PIPE_ALT_CLK_REQ	Alternate Clock Source Request. Indicates that the alternate clocks are needed by the slave PCS (that is, to boot the master MPLL). Connect to the alt_clk_en on the master.
0	PIPE_ALT_CLK_EN	Alternate Clock Enable. Enables the ref_pcs_clk and ref_pipe_pclk output clocks (if necessary, powers up the MPLL).

3.3.3 USB_PHY_CTL2 Register [Offset = 0x0008]

Figure 3-122. USB_PHY_CTL2 Register

31 30	29 27	26 23	22 21
Reserved	PHY_PC_LOS_BIAS	PHY_PC_TXVREFTUNE	PHY_PC_TXRISETUNE
R-0	R/W-101	R/W-1000	R/W-01

20 19	18 17	16 15	14
PHY_PC_TXRESTUNE	PHY_PC_TXPREEMPPULSETUNE	PHY_PC_TXPREEMPAMPTUNE	PHY_PC_TXHSXVTUNE
R/W-01	R/W-0	R/W-00	R/W-11

13 10	9 7	6 4	3	2 0
PHY_PC_TXFSLSTUNE	PHY_PC_SQRXTUNE	PHY_PC_OTGTUNE	Reserved	PHY_PC_COMPDISTUNE
R/W-0011	R/W-011	R/W-100	R-0	R/W-100

Legend: R = Read only; R/W = Read/Write, - *n* = value after reset

Table 3-135. USB_PHY_CTL2 Register Field Descriptions

Bit	Field	Description
31-30	Reserved	Reserved
29-27	PHY_PC_LOS_BIAS	Loss-of-Signal Detector Threshold Level Control. Sets the LOS detection threshold level. • +1 = results in a +15 mVp incremental change in the LOS threshold. • -1 = results in a -15 mVp incremental change in the LOS threshold. Note: the 000b setting is reserved and must not be used.
26-23	PHY_PC_TXVREFTUNE	HS DC Voltage Level Adjustment. Adjusts the high-speed DC level voltage. • +1 = results in a +1.25% incremental change in high-speed DC voltage level. • -1 = results in a -1.25% incremental change in high-speed DC voltage level.
22-21	PHY_PC_TXRISETUNE	HS Transmitter Rise/Fall TIme Adjustment. Adjusts the rise/fall times of the high-speed waveform. • +1 = results in a -4% incremental change in the HS rise/fall time. • -1 = results in a +4% incremental change in the HS rise/fall time.
20-19	PHY_PC_TXRESTUNE	USB Source Impedance Adjustment. Some applications require additional devices to be added on the USB, such as a series switch, which can add significant series resistance. This bus adjusts the driver source impedance to compensate for added series resistance on the USB.
18	PHY_PC_TXPREEMPPULSETUNE	HS Transmitter Pre-Emphasis Duration Control. Controls the duration for which the HS pre-emphasis current is sourced onto DP or DM. It is defined in terms of unit amounts. One unit of pre-emphasis duration is approximately 580 ps and is defined as 1x pre-emphasis duration. This signal valid only if either txpreempamptune[1] or txpreempamptune[0] is set to 1. • 1 = 1x, short pre-emphasis current duration. • 0 = 2x, long pre-emphasis current duration.

Table 3-135. USB_PHY_CTL2 Register Field Descriptions (continued)

Bit	Field	Description
17-16	PHY_PC_TXPREEMPAMPTUNE	HS Transmitter Pre-Emphasis Current Control. Controls the amount of current sourced to DP and DM after a J-to-K or K-to-J transition. The HS Transmitter pre-emphasis current is defined in terms of unit amounts. One unit amount is approximately 600 uA and is defined as 1x pre-emphasis current. • 11 = 3x pre-emphasis current. • 10 = 2x pre-emphasis current. • 01 = 1x pre-emphasis current. • 00 = HS Transmitter pre-emphasis is disabled.
15-14	PHY_PC_TXHSXVTUNE	Transmitter High-Speed Crossover Adjustment. Adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode. • 11 = Default setting. • 10 = +15 mV • 01 = -15 mV • 00 = Reserved
13-10	PHY_PC_TXFSLSTUNE	FS/LS Source Impedance Adjustment. Adjusts the low- and full-speed single-ended source impedance while driving high. This parameter control is encoded in thermometer code. • +1 = results in a -2.5% incremental change in threshold voltage level. • -1 = results in a +2.5% incremental change in threshold voltage level. Any non-thermometer code setting (that is 1001) is not supported and reserved.
9-7	PHY_PC_SQRXTUNE	Squelch Threshold Adjustment. Adjusts the voltage level for the threshold used to detect valid high-speed data. • +1 = results in a -5% incremental change in threshold voltage level. • -1 = results in a +5% incremental change in threshold voltage level.
6-4	PHY_PC_OTGTUNE	VBUS Valid Threshold Adjustment. Adjusts the voltage level for the VBUS valid threshold. • +1 = results in a +1.5% incremental change in threshold voltage level. • -1 = results in a -1.5% incremental change in threshold voltage level.
3	Reserved	Reserved
2-0	PHY_PC_COMPDISTUNE	Disconnect Threshold Adjustment. Adjusts the voltage level for the threshold used to detect a disconnect event at the host. • +1 = results in a +1.5% incremental change in the threshold voltage level. • -1 = results in a -1.5% incremental change in the threshold voltage level.

3.3.4 USB_PHY_CTL3 Register [Offset = 0x000C]

Figure 3-123. USB_PHY_CTL3 Register

31	30	29	23
Reserved		PHY_PC_PCS_TX_SWING_FULL	
R-0		R/W-1111000	

22	17	16	11	10	5	4	0
PHY_PC_PCS_TX_DEEMPH_6DB		Reserved		PHY_PC_PCS_TX_DEEMPH_3P5DB		PHY_PC_LOS_LEVEL	
R/W-100000		R-0		R/W-010101		R/W-01001	

Legend: R = Read only; R/W = Read/Write, - *n* = value after reset

Table 3-136. USB_PHY_CTL3 Register Field Descriptions

Bit	Field	Description
31-30	Reserved	Reserved
29-23	PHY_PC_PCS_TX_SWING_FULL	Tx Amplitude (Full Swing Mode). Sets the launch amplitude of the transmitter. It can be used to tune Rx eye for compliance.

Table 3-136. USB_PHY_CTL3 Register Field Descriptions (continued)

Bit	Field	Description
22-17	PHY_PC_PCS_TX_DEEMPH_6DB	Tx De-Emphasis at 6 dB. Sets the Tx driver de-emphasis value when pipeP_tx_deemph[1:0] is set to 10b (according to the PIPE3 specification). This bus is provided for completeness and as a second potential launch amplitude.
16-11	Reserved	Reserved
10-5	PHY_PC_PCS_TX_DEEMPH_3P5DB	Tx De-Emphasis at 3.5 dB. Sets the Tx driver de-emphasis value when pipeP_tx_deemph[1:0] is set to 10b (according to the PIPE3 specification). Can be used for Rx eye compliance.
4-0	PHY_PC_LOS_LEVEL	Loss-of-Signal Detector Sensitivity Level Control. Sets the LOS detection threshold level. This signal must be set to 0x9.

3.3.5 USB_PHY_CTL4 Register [Offset = 0x0010]

Figure 3-124. USB_PHY_CTL4 Register

31	30	29	28
PHY_SSC_EN	PHY_REF_USE_PAD	PHY_REF_SSP_EN	PHY_MPLL_REFSSC_CLK_EN
R/W-1	R/W-1	R/W-0	R/W-0

27 22	21	20 19	18	17
PHY_FSEL	PHY_RETENABLEN	PHY_REFCLKSEL	PHY_COMMONONN	CTRL_MISC_DEBUG_EN
R/W-100111	R/W-1	R/W-10	R/W-0	R-0

16	15	14 12	11 7	6 0
PHY_OTG_VBUSVLDEXTSEL	PHY_OTG_OTGDISABLE	PHY_PC_TX_VBOOST_LVL	PHY_PC_LANE0_TX_TERM_OFFSET	Reserved
R/W-0	R/W-1	R/W-100	R/W-00000	R-0

Legend: R = Read only; R/W = Read/Write, - *n* = value after reset

Table 3-137. USB_PHY_CTL4 Register Field Descriptions

Bit	Field	Description
31	PHY_SSC_EN	Spread Spectrum Enable. Enables spread spectrum clock production (0.5% down-spread at ~31.5 KHz) in the USB3.0 PHY. If the reference clock already has spread spectrum applied, ssc_en must be de-asserted.
30	PHY_REF_USE_PAD	Select Reference Clock Connected to ref_pad_clk_{p,m}. When asserted, selects the external ref_pad_clk_{p,m} inputs as the reference clock source. When de-asserted, ref_alt_clk_{p,m} are selected for an on-chip reference clock source. Note: the default value is 0 on K2E and K2L devices.
29	PHY_REF_SSP_EN	Reference Clock Enables for SS function. Enables the reference clock to the prescaler. The ref_ssp_en signal must remain de asserted until the reference clock is running at the appropriate frequency, at which point ref_ssp_en can be asserted. For lower power states, ref_ssp_en can also be de asserted.
28	PHY_MPLL_REFSSC_CLK_EN	Double-Word Clock Enable. Enables/disables the mpll_refssc_clk signal. To prevent clock glitch, it must be changed when the PHY is inactive.
27-22	PHY_FSEL	Frequency Selection. Selects the reference clock frequency used for both SS and HS operations. The value for fsel combined with the other clock and enable signals will determine the clock frequency used for SS and HS operations and if a shared or separate reference clock will be used.
21	PHY_RETENABLEN	Lowered Digital Supply Indicator. Indicates that the vp digital power supply has been lowered in Suspend mode. This signal must be de-asserted before the digital power supply is lowered. • 1 = Normal operating mode. • 0 = The analog blocks are powered down.

Table 3-137. USB_PHY_CTL4 Register Field Descriptions (continued)

Bit	Field	Description
20-19	PHY_REFCLKSEL	Reference Clock Select for PLL Block. Selects reference clock source for the HS PLL block. • 11 = HS PLL uses EXTREFCLK as reference. • 10 = HS PLL uses either ref_pad_clk_{p,m} or ref_alt_clk_{p,m} as reference. • x0 = Reserved.
18	PHY_COMMONONN	Common Block Power-Down Control. Controls the power-down signals in the HS Bias and PLL blocks when the USB3.0 PHY is in Suspend or Sleep mode. • 1 = In Suspend or Sleep mode, the HS Bias and PLL blocks are powered down. • 0 = In Suspend or Sleep mode, the HS Bias and PLL blocks remain powered and continue to draw current.
17	CTRL_MISC_DEBUG_EN	Defined on K2E and K2L devices, but not used on K2K and K2H devices.
16	PHY_OTG_VBUSVLDEXTSEL	External VBUS Valid Select. Selects the VBUSVLDEXT input or the internal Session Valid comparator to indicate when the VBUS signal on the USB cable is valid. • 1 = VBUSVLDEXT input is used. • 0 = Internal Session Valid comparator is used.
15	PHY_OTG_OTGDISABLE	OTG Block Disable. Powers down the OTG block, which disables the VBUS Valid and Session End comparators. The Session Valid comparator (the output of which is used to enable the pull-up resistor on DP in Device mode) is always on irrespective of the state of otgdisable. If the application does not use the OTG function, setting this signal to high to save power. • 1 = OTG block is powered down. • 0 = OTG block is powered up.
14-12	PHY_PC_TX_VBOOST_LVL	Tx Voltage Boost Level. Sets the boosted transmit launch amplitude (mV_{ppd}). The default setting is intended to set the launch amplitude to approximately 1,008mV_{ppd}. • +1 = results in a +156 mV_{ppd} change in the Tx launch amplitude. • -1 = results in a -156 mV_{ppd} change in the Tx launch amplitude.
11-7	PHY_PC_LANE0_TX_TERM_OFFSET	Transmitter Termination Offset. Enables adjusting the transmitter termination value from the default value of 60 Ω.
6-0	Reserved	

3.3.6 USB_PHY_CTL5 Register [Offset = 0x0014]

Figure 3-125. USB_PHY_CTL5 Register

31	21	20	19	13
Reserved		PHY_REF_CLKDIV2	PHY_MPLL_MULTIPLIER[6:0]	
R-0		R/W-0	R/W +0011001	

12	4	3	2	0
PHY_SSC_REF_CLK_SEL		Reserved	PHY_SSC_RANGE	
R/W-000000000		R-0	R/W-000	

Legend: R = Read only; R/W = Read/Write, - *n* = value after reset

Table 3-138. USB_PHY_CTL5 Register Field Descriptions

Bit	Field	Description
31-21	Reserved	Reserved
20	PHY_REF_CLKDIV2	Input Reference CLock Divider Control. If the input reference clock frequency is greater than 100MHz, this signal must be asserted. The reference clock frequency is then divided by 2 to keep it in the range required by the MPLL. When this input is asserted, the ref_ana_usb2_clk (if used) frequency will be the reference clock frequency divided by 4.
19-13	PHY_MPLL_MULTIPLIER[6:0]	MPLL Frequency Multiplier Control. Multiplies the reference clock to a frequency suitable for intended operating speed.
12-4	PHY_SSC_REF_CLK_SEL	Spread Spectrum Reference Clock Shifting. Enables non-standard oscillator frequencies to generate targeted MPLL output rates. Input corresponds to frequency-synthesis coefficient. • . ssc_ref_clk_sel[8:6] = modulous - 1 • . ssc_ref_clk_sel[5:0] = 2's complement push amount.
3	Reserved	Reserved
2-0	PHY_SSC_RANGE	Spread Spectrum CLock Range. Selects the range of spread spectrum modulation when ssc_en is asserted and the PHY is spreading the high-speed transmit clocks. Applies a fixed offset to the phase accumulator.

Supplemental Information

This appendix contains supplementary information.

Topic **Page**

A.1 Section 1 LTSSM State / Substat Encoding

Table A-2 and Table A-2 shows the encoding used for the link training SuperSpeed state machine (LTSSM) state (ltdb_link_state, 4 bit) and substate (ltdb_substate, 4 bit), which is available in the core observability mode as shown in Table 2-63. The state and substate are also visible to the software in the USB_GDBGLTSSM register. The substate can only be decoded relatively to the state (that is, all 8 bits must be decoded together). There are more substates in the current implementation than there are LTSSM substates defined by the standard (see *Universal Serial Bus Revision 3.0 Specification)*, because the implementation is finer grained than the standard. Table A-2 describes the relationship.

Table A-1. LTSSM State Encoding For core_ltdb_link_state[3:0]

Encoding	LTSSM State Name
0x0	U0
0x1	U1
0x2	U2
0x3	U3
0x4	SS.Disabled
0x5	Rx.Detect
0x6	SS.Inactive
0x7	Polling
0x8	Recovery
0x9	Hot Reset
0xA	Compliance Mode
0xB	Loopback

Table A-2. LTSSM State/Substate Encoding

core_ltdb_link_state[3:0]	core_ltdb_substate[3:0]	LTSSM State	LTSSM Substate	Implementation-Specific Substate Name	Description
0x0	x	U0	N/A	U0	U0: No substate
0x1	0x0	U1		U1_POWER	PHY power state P0-P1 request
0x1	0x1	U1		U1_POWER_A	Wait for PHY power change done
0x1	0x2	U1		U1_ACTIVE	Wait for remote/local U1 exit
0x1	0x3	U1		U1_EXIT_LOC_RESP	Start U1 exit and wait for min response from remote partner
0x1	0x4	U1		U1_EXIT_LOC_FIN	Locally initiated U1 exit
0x1	0x5	U1		U1_EXIT_REM	Remote initiated U1 exit
0x1	0x6	U1		U1_EXIT_P0	PHY power state P1-P0 request
0x1	0x7	U1		U1_EXIT_P0_A	Wait for PHY power change P1-P0 done
0x1	0x8	U1		U1_EXIT_DONE	U1 exit successful U1-Recovery
0x2	0x0	U2		U2_POWER	PHY power state P0-P2 request
0x2	0x1	U2		U2_POWER2	Wait for PHY power change done
0x2	0x2	U2		U2_ACTIVE	Wait for remote/local U2 exit
0x2	0x3	U2		U2_ACTIVE_0	Request to start receiver detection
0x2	0x4	U2		U2_ACTIVE_1	Wait for receiver detection response
0x2	0x5	U2		U2_EXIT_LOC_RESP	Start U2 exit and wait for minimum response from remote partner
0x2	0x6	U2		U2_EXIT_LOC_FIN	Locally initiated U2 exit
0x2	0x7	U2		U2_EXIT_REM	Remote initiated U2 exit
0x2	0x8	U2		U2_EXIT_P0	PHY power state P2–P0 request

Table A-2. LTSSM State/Substate Encoding (continued)

core_ltdb_link_state[3:0]	core_ltdb_substate[3:0]	LTSSM State	LTSSM Substate	Implementation-Specific Substate Name	Description
0x2	0x9	U2		U2_EXIT_P0_A	Wait for PHY power change P1-P0 done
0x2	0xA	U2		U2_EXIT_DONE	U2 exit successful U2-Recovery
0x3	0x0	U3		U3_POWER	PHY power state P0-P2/P3 request
0x3	0x1	U3		U3_POWER3	Wait for PHY power change done
0x3	0x2	U3		U3_ACTIVE	Wait for remote/local U3 exit
0x3	0x3	U3		U3_ACTIVE_0	Request to start receiver detection
0x3	0x4	U3		U3_ACTIVE_1	Wait for receiver detection response
0x3	0x5	U3		U3_EXIT_LOC_POW	PHY power state P3-P0-P2 request
0x3	0x6	U3		U3_EXIT_LOC_POW_A	Wait for PHY power change done
0x3	0x7	U3		U3_EXIT_LOC_RESP	Start U3 exit and wait for minimum response from remote partner
0x3	0x8	U3		U3_EXIT_LOC_FIN	Locally initiated U3 exit
0x3	0x9	U3		U3_EXIT_REM_POW	PHY Power state P3-P0-P2 request
0x3	0xA	U3		U3_EXIT_REM_POW_A	Wait for PHY power change done
0x3	0xB	U3		U3_EXIT_REM	Remote initiated U3 exit
0x3	0xC	U3		U3_EXIT_P0	PHY power state P2/P3-P0 request
0x3	0xD	U3		U3_EXIT_P0_A	Wait for PHY power change P2/P3-P0 done
0x3	0xE	U3		U3_EXIT	Received remote/local U3 exit
0x3	0xF	U3		U3_EXIT_DONE	U3 exit successfu lU3-Recovery
0x4	0x0	SS.Disabled		SSD_POWER3	PHY power state P2/P3 request
0x4	0x1	SS.Disabled		SSD_POWER3A	Wait for PHY power change done
0x4	0x2	SS.Disabled		SSD_MAIN	Wait for RUN/STOP bit
0x5	0x0	Rx.Detect		RXD_INIT	PHY power state P2 request
0x5	0x1	Rx.Detect		RXD_POWER2	Wait for PHY power change done
0x5	0x2	Rx.Detect	Reset	RXD_RESET	Warm reset
0x5	0x3	Rx.Detect	Reset	RXD_RESET_T	Wait for RUN/STOP bit (device mode)
0x5	0x4	Rx.Detect	Active	RXD_ACTIVE0	Request to start receiver detection
0x5	0x5	Rx.Detect	Active	RXD_ACTIVE1	Wait for receiver detection response
0x5	0x6	Rx.Detect	Quiet	RXD_QUIET	wait for 12-ms timer timeout
0x6	0x0	SS.Inactive		SSI_RESET	PHY power state P0-P2 request
0x6	0x1	SS.Inactive		SSI_POWER2	Wait for PHY power change done
0x6	0x2	SS.Inactive	Quiet	SSI_QUIET0	Start 12-ms timer
0x6	0x3	SS.Inactive	Quiet	SSI_QUIET1	Wait for 12-ms timer timeout
0x6	0x4	SS.Inactive	Disconnect.Detect	SSI_DIS_DET0	Request to start receiver detection
0x6	0x5	SS.Inactive	Disconnect.Detect	SSI_DIS_DET1	Wait for receiver detection response
0x7	0x0	Polling		POLL_RESET	PHY power state P0 request
0x7	0x1	Polling		POLL_POWER0	Wait for PHY power change done
0x7	0x2	Polling	LFPS	POLL_LFPS	Send/receive Poll.LFPS
0x7	0x3	Polling	RxEQ	POLL_RXEQ	Send/receive TSEQ training set
0x7	0x4	Polling	Active	POLL_ACTIVE	Send/receive TS1 training set
0x7	0x5	Polling	Configuration	POLL_CONFIG	Send/receive TS2 training set

Table A-2. LTSSM State/Substate Encoding (continued)

core_ltdb_link_state[3:0]	core_ltdb_substate[3:0]	LTSSM State	LTSSM Substate	Implementation-Specific Substate Name	Description
0x7	0x6	Polling	Idle	POLL_IDLE	Send/receive logical IDLE symbols
0x8	0x0	Recovery		RECOV_RESET	PHY power state P0 request
0x8	0x1	Recovery		RECOV_POWER0	Wait for PHY power change done
0x8	0x2	Recovery	Active	RECOV_ACTIVE	Send/receive TS1 training set
0x8	0x3	Recovery	Configuration	RECOV_CONFIG	Send/receive TS2 training set
0x8	0x4	Recovery	Idle	RECOV_IDLE	Send/receive logical IDLE symbols
0x9	0x0	Hot Reset		HRESET_RST	Request TCRRL to send TS2
0x9	0x1	Hot Reset		HRESET_GO	Start 12-ms timer
0x9	0x2	Hot Reset	Active	HRESET_ACT1	Send/receive TS1 training set with reset bit
0x9	0x3	Hot Reset	Active	HRESET_ACT2	Send/receive TS1 training set without reset bit
0x9	0x4	Hot Reset	Exit	HRESET_EXIT	Send/receive logical IDLE symbols
0xA	x	Compliance Mode	N/A	N/A	N/A
0xB	0x0	Loopback	Active	LPBK_IDLE	Wait for loopback start request
0xB	0x1	Loopback	Active	LPBK_MASTER	Master mode: Wait for loopback exit request
0xB	0x2	Loopback	Active	LPBK_SLAVE	Slave mode: Wait for loopback exit request
0xB	0x3	Loopback	Exit	LPBK_EXIT_LOC_RESP	Start loopback exit and wait for minimum response from remote partner
0xB	0x4	Loopback	Exit	LPBK_EXIT_LOC_FIN	Locally initiated loopback exit
0xB	0x5	Loopback	Exit	LPBK_EXIT_REM	Remote initiated loopback exit

Table A-3 to Table A-6 show the encoding used for various, nonstandard core observability fields.

The USB2.0 port described in Table A-3 can be either in HS, FS, or LS mode, depending on the configuration and on the attached USB device.

Table A-3. USB2.0 Port State Encoding

core_u2_prt_state[4:0]	USB2.0 Port State
0x00	PRT_DISCON
0x01	DISABLE
0x02	FSLS_RESET
0x03	WAIT_CHIRP
0x04	Reserved
0x05	DEV_CHIRP
0x06	HST_CHIRP_J
0x07	HST_CHIRP_K
0x08	HST_CHIRP_END
0x09	ENABLE
0x0A	AEOF
0x0B	SOF
0x0C	SUSPEND
0x0D	RESUME
0x0E	RESUME_END
0x0F	RESUME_DONE

Table A-3. USB2.0 Port State Encoding (continued)

core_u2_prt_state[4:0]	USB2.0 Port State
0x10	TEST_PKT
0x11	PRE_ENABLE
0x12	SOF_END
0x13	TEST
0x14	Reserved
0x15	HST_CHIRP_DONE

The USB controller includes two USB2.0 media access controller (MAC) instances. Table A-4 describes their state. In the current single-port architecture, only one can be operational at a time.

- MAC 0 (core_u2mac_txrx_state_0[4:0]) is the FS/LS instance.
- MAC 1 (core_u2mac_txrx_state_1[4:0]) is the HS instance.

Table A-4. USB2.0 MAC State Encoding

core_u2mac_txrx_state_X[4:0]	USB2.0 MAC State	Note
0x00	IDLE	
0x01	SEND_SPLT	Host mode only
0x02	TKN2TKN	Host mode only
0x03	SEND_CRC5	Host mode only
0x04	TKN2DAT	Host mode only
0x05	WAIT_DPKT	Device mode only
0x06	RECV_ERR	
0x07	RECV_DATA	
0x08	SEND_HSHK	
0x09	SEND_EOP	
0x0A	SEND_DATA	
0x0B	WAIT_HSHK	
0x0C	CHK_CRC16	
0x0D	WAIT_DATA	
0x0E	TESTMOD	Device mode only
0x0F	WAIT_BUSIDLE	
0x10	WAIT_TURNAROUND	Device mode only
0x11	DATA_TO_STS	
0x12	WAIT_EOP	Host mode only

The USB controller includes a 4-bit device speed/suspend/resume (DSSR) submodule, the state of which is described in Table A-5.

Table A-5. USB2.0 DSSR State Encoding

core_u2_dssr_state[3:0]	USB2.0 Port State
0x00	DEV_INIT
0x01	DEV_IDLE
0x02	PULLUP_EN
0x03	CHK_BUS
0x04	SEND_CHIRP
0x05	WAIT_HST_CHIRP
0x06	SUSPENDED
0x07	RESUMING

Table A-5. USB2.0 DSSR State Encoding (continued)

core_u2_dssr_state[3:0]	USB2.0 Port State
0x08	FLS_RESET
0x09	DLINE_PULSING
0x0A	PULSING_DONE
0x0B	DEV_CHIRP_END
0x0C	DEV_CHIRP_DONE
0x0D	REMOTE_WAKEUP
0x0E	RMWAKEUP_DONE

Table A-6 shows the current USB role encoding.

Table A-6. Current Mode Encoding

core_sm2bl_cur_mode	Current USB Mode (Role)
0x0	USB host
0x1	USB device

Revision A History

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

Changes from Original (April 2014) to A Revision	Page
• Changed instances of *(SuperSpeed)* to *(USB 2.0)* in Table 3-48	94
• Changed instances of *(USB2.0)* to *(SuperSpeed)* in Table 3-48	94

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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