KeyStone Architecture
HyperLink

User Guide

Literature Number: SPRUGW8C
June 2013

Release History

Release	Date	Description/Comments
SPRUGW8C	June 2013	• Corrected pin number from 24 pins to 26 pins (Page 1-2) • Made corrections to the Address Translation at RX Side graphic. (Page 2-22) • Fixed minor typos (Page 2-22) • Corrected the segment size (Page 2-23) • Corrected typo in the formula (Page 2-23) • Fixed typo in the register chapter 3.2.9 (Page 3-11)
SPRUGW8B	October 2012	• Added Side band state register for Keystone II Hyperlink (Page 3-17) • Added Rx Priority Control register for Keystone II devices (Page 3-17) • Added Rx Priority Control register for Keystone II devices (Page 3-15) • Added RTL version and revmin for Keystone II devices (Page 3-4) • Added section HyperLink SerDes Training Process (Page 2-11) • Updated SERDES programming chapter to enable PLL after running all Tx, Rx configuration (Page A-2) • Add Link Status Register in the index table (Page 3-2) • Added link status register in the register map table (Page 3-2) • Modified several register descriptions for clarification (Page 3-4)
SPRUGW8A	November 2011	• Added the appendix "Programming Examples" (Page A-1) • Updated SerDes Configuration and Clocking to add details about the SerDes operating rate, receive interface, and transmit interface (Page 2-8) • Updated registers chapter to add the HyperLink SerDes configuration and status registers. (Page 3-1) • Updated sections in the register chapter to distinguish between Local registers, Remote registers, and SerDes registers. (Page 3-1) • Changed BUSWIDTH field description (Page 3-26)
SPRUGW8	November 2010	Initial Release

Contents

Chapter 3

Appendix A

List of Tables

List of Figures

Preface

About This Manual

HyperLink provides a high-speed, low-latency, and low-pin-count communication interface that extends the internal CBA 3.x-based transactions between two KeyStone devices.

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

C66x CorePac User Guide	SPRUGW0
Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User Guide	SPRUGS5
Chip Interrupt Controller (CIC) for KeyStone Devices User Guide	SPRUGW4
Memory Protection Unit (MPU) for KeyStone Devices User Guide	SPRUGW5
Multicore Navigator for KeyStone Devices User Guide	SPRUGR9
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

Trademarks

C6000 is a trademark of Texas Instruments Incorporated.

All other brand names and trademarks mentioned in this document are the property of Texas Instruments Incorporated or their respective owners, as applicable.

Chapter 1

Introduction

1.1 Purpose of HyperLink

HyperLink provides a high-speed, low-latency, and low-pin-count communication interface that extends the internal CBA 3.x-based transactions between two KeyStone devices. It can emulate all currently-used peripheral interface mechanisms. HyperLink includes the data signals and the sideband control signals. The data signals are SerDes-based and the sideband control signals are LVCMOS-based. The current version of HyperLink offers point-to-point connection between two devices.

1.2 Terminology Used in This Document

Term	Definition
CBA	Common Bus Architecture
LVCMOS	Low Voltage Complementary Metal Oxide Semiconductor
QM	Queue Manager
SerDes	Serializer/Deserializer

1.3 Features

The HyperLink module has the following features:

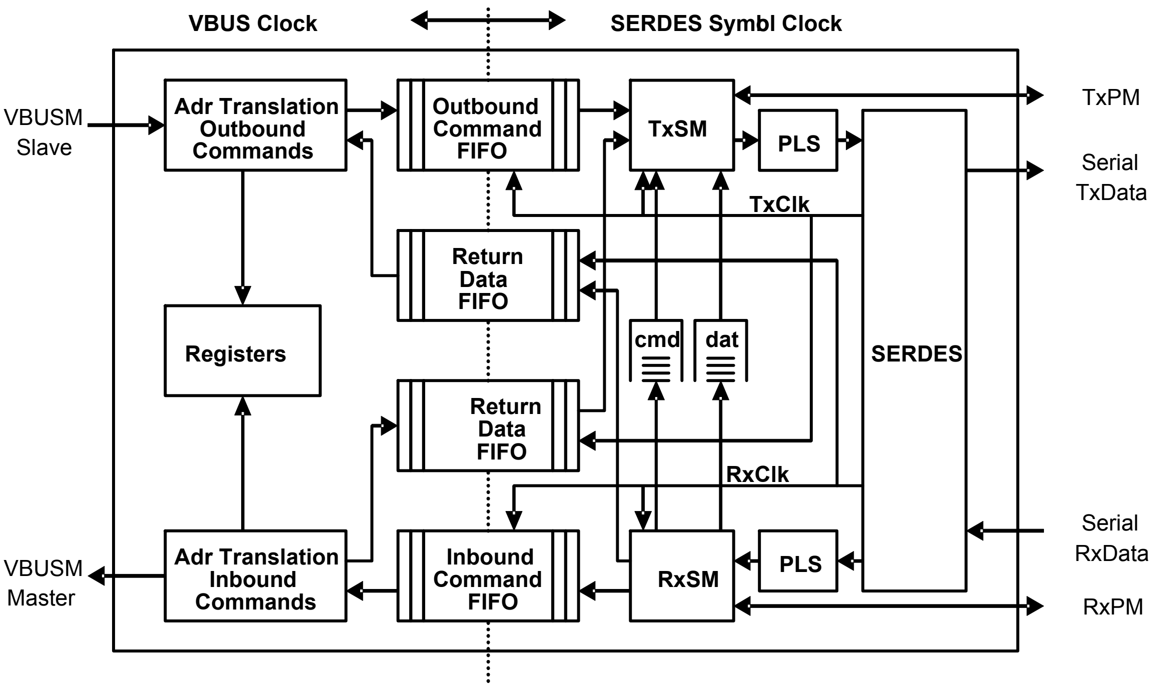
- Low pin count (as few as 26 pins)
  - SerDes for data transfer
  - LVCMOS sideband signals dedicated for control
- No Tri-State Signals
  - All signals are dedicated and driven by only one device
  - All LVCMOS sideband signals are driven using source synchronous clocking
- Up to 12.5 Gbaud rate per lane, 1 or 4 lanes for Tx and Rx data
  - Support SerDes full, half, quarter and eighth rates for slower speeds
  - Auto SerDes polarity detection and correction
  - Auto SerDes lane identification and correction
- Simple packet-based transfer protocol for memory-mapped access
  - Write Request/Data Packet
  - Read-request packet
  - Read-response data packet
  - Interrupt-request packet
  - Supports multiple outstanding transactions
- Point-to-point connection
  - Request packets and response packets are multiplexed through the same physical pins
  - Supports both host/peripheral and peer-to-peer communication models
- Dedicated LVCMOS pins for flow control and power management
  - Supports per-direction, per-channel flow control
  - Supports per-lane, per-direction power management
- Automatically adjusts lane width for power saving
- Internal SerDes loopback mode for diagnostics
- Requires no external pull-up or pull-down resistors
- 64 interrupt inputs for both hardware and software

- Eight interrupt-pointer addresses
- Does not support write-response packets
- Both TX and RX SerDes must run at the same speed
- Extended control word for commands not used in this version
- Maximum burst size supported to access remote registers is 64 bytes. Bursts greater than 64B may cause CBA violations.
- No exclusive transport operation supported
- CBA constant mode is not supported for bursts larger than 256-byte aligned burst

1.4 Functional Block Diagram

Figure 1-1 shows the internal modules of the HyperLink block.

Figure 1-1 Block Diagram



The HyperLink module implements two 256-bit, VBUSM interfaces. The slave interface is required for transmit and control register access and the master interface is required for receive. Transmit and Receive state machine blocks convert to/from the 256-bit CBA bus to the external serial interface.

There are address transaction blocks at the ingress and egress sides. The egress address combines the various characteristics of the CBA transactions the HyperLink slave port receives—such as security features, privID information, and memory-mapped addresses—into the outgoing address. On the ingress side, the HyperLink takes the address fields and remaps them into a CBA transaction. The address translations for ingress and egress are done independently and, therefore, controlled by different sets of registers to increase processing flexibility and scalability.

Because HyperLink logic and SerDes may run different clocks, multiple FIFOs are added to buffer the entire data burst and multiple commands. Ingress and egress have their own FIFOs, as does the read-return data for ingress and egress.

The station management block processes the power management and flow control sideband signals, such as power a lane up/down, flow control, enable/disable TX/RX transceiver, etc. The station management is also used for initialization and error recovery.

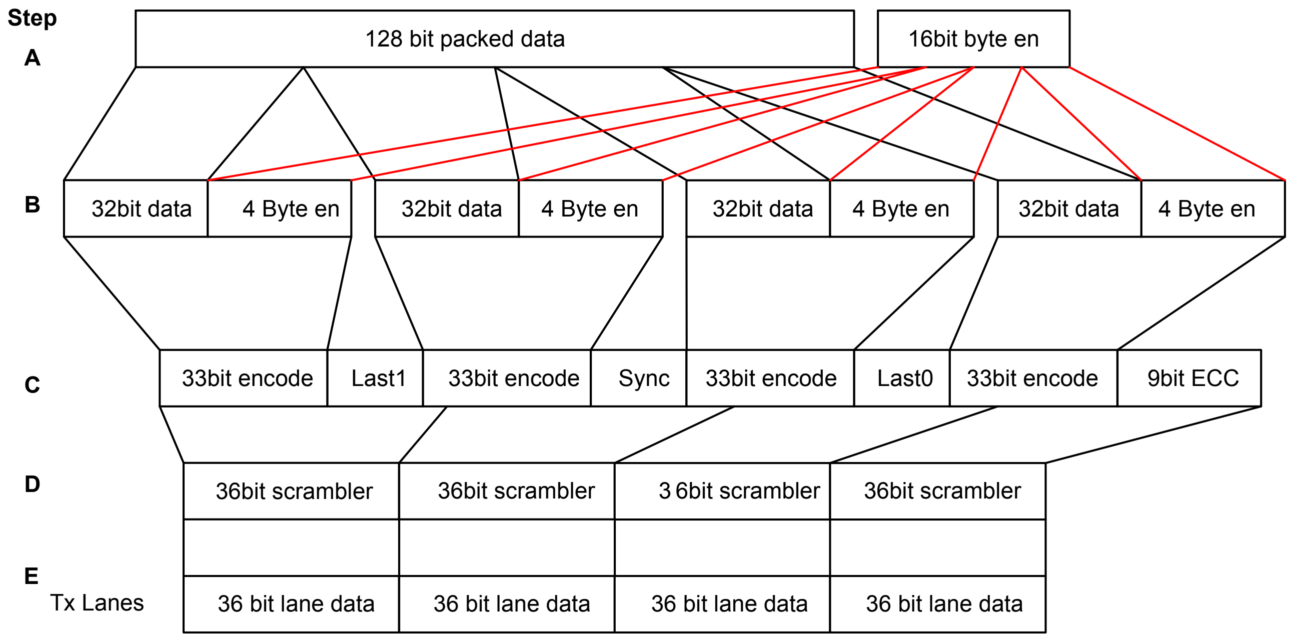
The PLS module on the egress side encodes the MAC transmit data using a GFP 32/33 encoding, adding a nine-bit Error Correction Code (ECC), then scrambling the data before sending it to the SerDes. On the ingress side, the PLS aligns the serial bit stream to the 36-bit symbol boundary, identifies the synchronizing codes, descrambles the data, aligns to the ECC boundary, performs the ECC correction, decodes data using GFP 33/32, and presents the resulting data to the MAC receiver.

When a command arrives at the slave interface for remote destination, a 64- or 128-bit command is written to the FIFO followed by any applicable data. Data is written in 8-, 16-, 24-, or 32-byte octet-aligned quanta. That is, the smallest write is eight bytes. The FIFO can accept up to 32 bytes of data per bus clock.

Figure 1-2 shows the flow of data from the FIFOs through the MAC and PLS to the wire.

- Step A: Takes up to 128 bits of data and associated byte enables (two-frame quanta) from the FIFO and slices it into 32-bit quanta with 4-bit byte enables. This results in 144 bits of information.
- Step B: Encodes the 32 bits of data and associated byte enables using a GFP encoding into 33 bits. This results in 132 bits of information.
- Step C: Takes four GFP encoded data words and adds the associated MAC last flags and a sync bit, calculates the ECC, and adds the nine bits of ECC to the word. This results in 144 bits of information.
- Step D: Splits the 144-bit word into four 36-bit lane data elements.
- Step E: Scrambles each lane to remove any repeating patterns that could cause a receive-recovery error.

Figure 1-2 MAC and PLS Operation



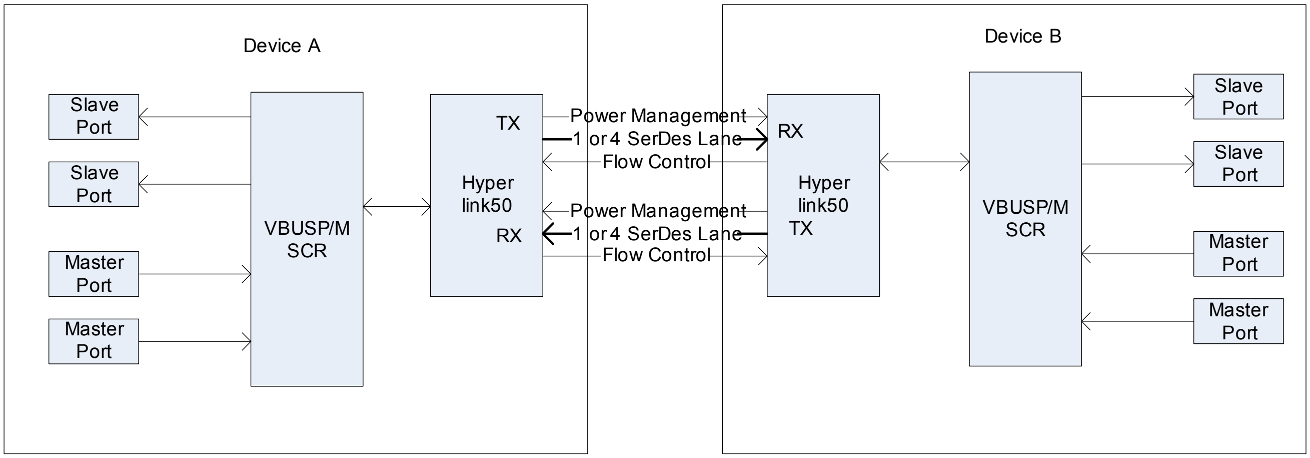
1.5 Supported Use Case Statement

HyperLink offers a packet-based transfer protocol and supports multiple outstanding read, write, and interrupt transactions. It can run in one-lane or four-lanes mode with each lane running at 12.5Gbaud rate. HyperLink uses an efficient encoding scheme for the physical layer. Compared to the traditional 8b10b encoding scheme for high speed interfaces, HyperLink reduces the encoding overhead; the efficient encoding scheme in HyperLink is equivalent to 8b9b.

The sideband signals provide the flow control and power management control information. After configuration, HyperLink has internal state machines to automatically manage the flow control and power saving based on the sideband signals without any software intervention. Flow control is managed independently on a per-direction basis. The RX side sends the throttle signals to the TX side. On the other hand, the power management is controlled by the TX side. In addition, power management is controlled on a per-lane, per-direction basis.

HyperLink is not compatible with VLYNQ due to the SerDes interface signaling.

Figure 1-3 Point-to-Point Connection Through HyperLink



1.6 Industry Standard(s) Compliance Statement

The HyperLink is a TI-specific peripheral. There is no industry standard for it.

Chapter 2

Architecture

2.1 Clock Control

HyperLink needs a reference clock for its SerDes module. The reference clock can be of the following frequencies:

- 156.25 Mhz
- 250 MHz
- 312 MHz

2.2 Memory Map

Not applicable.

2.3 Signal Descriptions

This section describes each of the signals on the external interface and their functions.

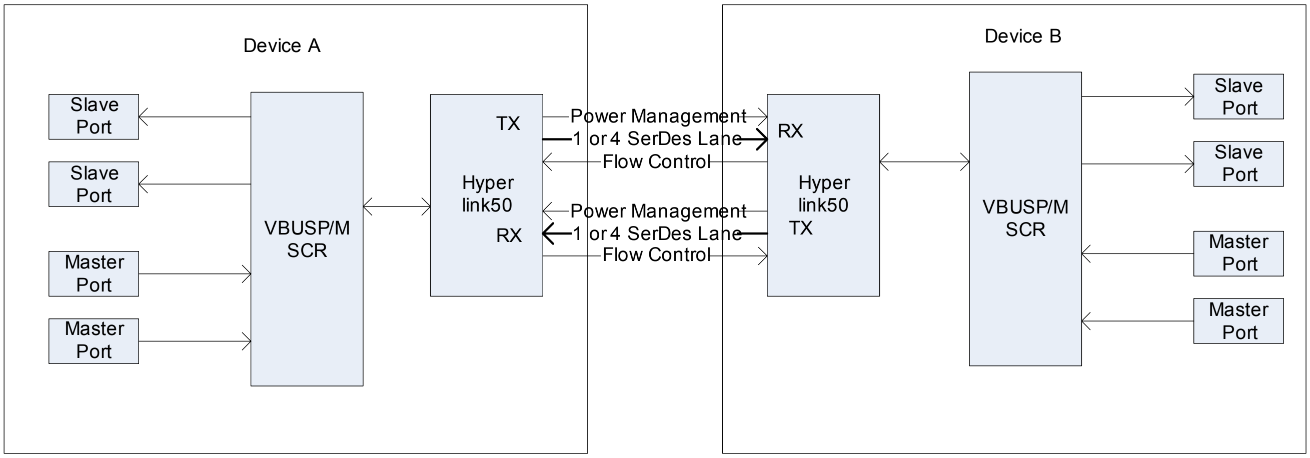
HyperLink includes both SerDes pins and LVCMOS pins. The SerDes pins are for data transfer and the LVCMOS pins are sideband signals for control, such as flow control and power management.

Table 2-1 HyperLink Pin Description

Pin Name	Pin Count	Type	Function
LVCMOS Pins			
TXPM_CLK_O	2	Out	Clock for transmit Power Management Output two wire bus
TXPM_DAT_O	2	Out	Transmit Power Management Output two wire bus
TXFL_CLK_I	2	In	Clock for transmit Flow Management Input two wire bus
TXFL_DAT_I	2	In	Transmit Flow Management Input two wire bus
RXPM_CLK_I	2	In	Received clock for receive Power Management Input two wire bus
RXPM_DAT_I	2	In	Receive Power Management Input two wire bus
RXFL_CLK_O	2	Out	Clock for receive Flow Management Output two wire bus
RXFL_DAT_O	2	Out	Receive Flow Management Output two wire bus
SerDes Pins			
SERDES_RXP0	1	In	Differential Rx pin lane 0 (Positive)
SERDES_RXN0	1	In	Differential Rx pin lane 0 (Negative)
SERDES_RXP1	1	In	Differential Rx pin lane 1(Positive)
SERDES_RXN1	1	In	Differential Rx pin lane 1 (Negative)
SERDES_RXP2	1	In	Differential Rx pin lane 2(Positive)
SERDES_RXN2	1	In	Differential Rx pin lane 2(Negative)
SERDES_RXP3	1	In	Differential Rx pin lane 3(Positive)
SERDES_RXN3	1	In	Differential Rx pin lane 3(Negative)
SERDES_REFCLKP	1	In	SerDes Differential Reference Clock (Positive)
SERDES_REFCLKN	1	In	SerDes Differential Reference Clock (Negative)

Figure 2-1 shows the connection between two devices through the HyperLink interface. Currently, HyperLink provides only point-to-point connections between two devices.

Figure 2-1 HyperLink Connection



2.4 Pin Multiplexing

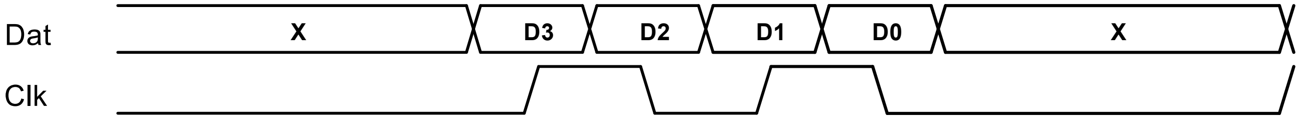
There is no pin multiplexing for HyperLink.

2.5 Station Management Interface

The station management interface refers to the LVCMOS pins in HyperLink, which are used for flow control and power management for HyperLink. The HyperLink module uses the TXPM and TXFL station management busses to control the SerDes TXDATA serial interface; the RXPM and RXFL station management busses are used to control the SerDes RXDATA serial interface.

Both flow control and power management have dedicated interfaces for sideband events.

Figure 2-2 Station Management Interface Timing



Each station management interface includes a data signal and clock. The data line carries a 4-bit value and is clocked by the receiver on each edge of the CLK signal. This methodology allows the clock and data to have the same maximum bandwidth, which enables the interface to potentially operate at higher rates for the same buffer type. The four-bit value in the data line supports up to 15 different messages which are sent MSB first. Message 0b1110 is used to transfer four bits of abilities followed by four bits of status. Bit 3 (MSB) of the status message is always zero to prevent the remote device from detecting a reset sequence.

To reset the station management bus, a sequence of eight 1s followed by four 0s is sent. Anytime the receiver of the station management bus sees more than seven 1s in a row, it resets its outputs to their default values and enters the sync state waiting for the four 0s. After the four 0s are received, the reset of the station management bus is complete.

When the station management bus is started, the transmit station management entity sends a reset sequence to the remote followed by a capabilities/status message. When a receive station management entity receives a reset sequence, it responds by sending a capabilities/status message; this allows either side of the link to start before the other and still determine remote capabilities. When both devices have received the capability/status message, the station management bus is considered linked and the capabilities are displayed in the status register.

The station management bus is clocked at one fourth the SerDes symbol rate or one fourth the VBUS_CLK rate.

The following tables show the commands sent on the TXPM or received on the RXPM bus. The capabilities are sent anytime reset is removed or when a reset sequence is seen on the receiver.

Table 2-2 Power Management Message

D3D2D1D0	Description
0b0000	Disable One Lane Receiver
0b0001	Enable one lane Receiver
0b0010	Disable Four Lane Receiver
0b0011	Enable Four Lane Receiver
0b0010 to 0b1101	Reserved
0b1110	Capabilities/Status Message
0b1111	Reserved for reset

Table 2-3 shows the commands sent RXFL or received on TXFL bus. The capabilities are sent anytime reset is removed or when a reset sequence is seen on the receiver.

Table 2-3 Flow Control Message

D3D2D1D0	Description
0b0000	Channel 0 Operational
0b0001	Channel 0 flow controlled
0b0010 to 0b0111	Reserved
0b1000	Receive synch not locked
0b1001	Receive sync locked
0b1010 to 0b1101	Reserved for Future Use
0b1110	Capabilities/Status Message
0b1111	Used for Reset

Each capability message contains four bits: D3D2D1D0. Only two bits are currently used. The following lists how to decode the capability message:

- D0 =1 means the high speed above 12.5 GBaud, D0=0 means support 12.5 GBaud and below.
- D1=1 means support four lane, D1=0 means it does not support four lane mode.
- D2 and D3 are currently reserved, should be always zero.

2.5.1 Lane Activation

To power up a lane, the transmitter informs the receiver of a power-up event through the TXPM. The receiver gets the power-up event through RXPM, then enables the receiver. When the receiver is fully synchronized with the transmitter training sequence, the receiver sends an event back to the transmitter to inform the transmitter that it can use the lane for data transfer. The receiver then detects the transition from the training sequence to the data receiving mode.

To support the symbol alignment within SerDes, an enable from the PLS needs to indicate to SerDes that symbol alignment can occur. The synchronization word described earlier contains a comma so that this can easily be done in SerDes. Each lane needs a symbol alignment and, depending on the state of the other lanes, phase correction may occur on all active lanes. This feature allows one operating lane to expand to four lanes without the need for the operating lane to be taken down during alignment with the other lanes.

Figure 2-3 shows a high-level timing of lane 0 startup.

Figure 2-3 Lane 0 Power Up Timing

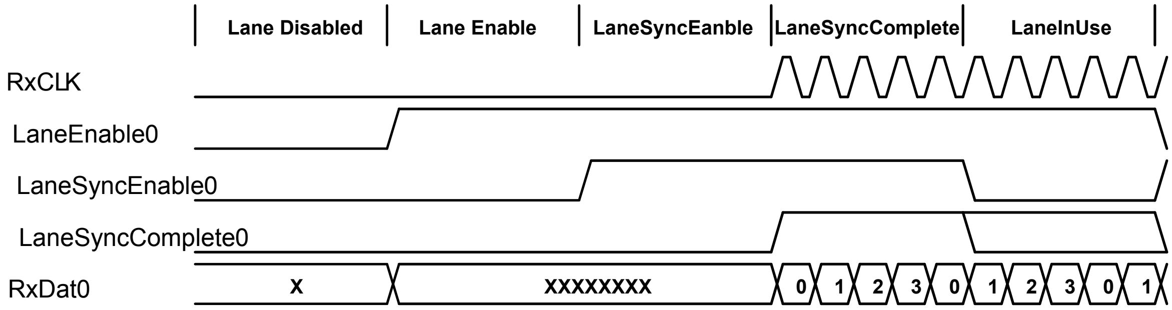
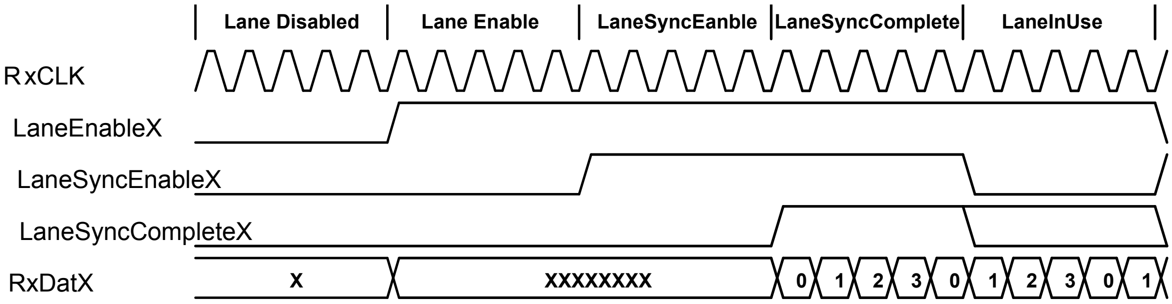


Figure 2-4 Timing Waveform for Second and Higher Lane Power Up.



Lane Disable—The SerDes lane is in its lowest power state. If all lanes are disabled, the SerDes power should be as close to zero as possible.

Lane Enable—The SerDes lane is enabled and the receiver is passing data seen on the Rx pins to the PLS layer.

LaneSyncEnable—SerDes is looking for the comma characters and, once they are detected, will symbol-align the land. SerDes will then set the LansSyncComplete back to the PLS for completion of the activation process.

LaneInUse—SerDes is aligned to the 36-bit word boundary. It ignores any comma characters detected. The PLS receives the DataInUse indicator and starts using the lane.

2.5.2 Flow Control

Flow control in the HyperLink module is transparent to the user. The RX side of HyperLink automatically manages the traffic flow based on available resources and throttles the TX side of traffic through the sideband signals.

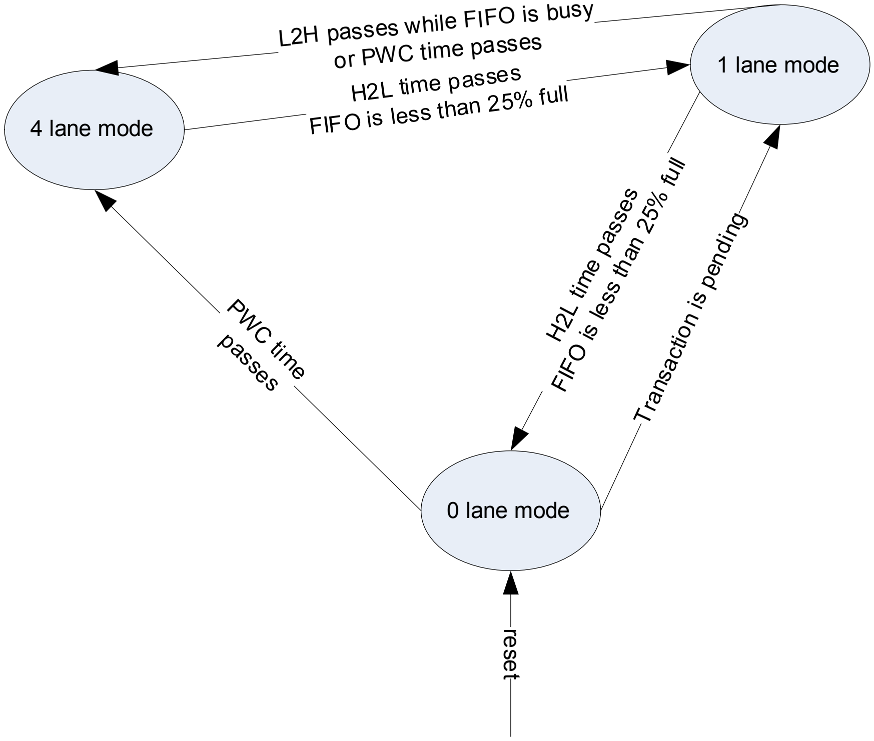
2.5.3 Power Management

The HyperLink transmitter actively determines the power states it enters based on the Lane Power Management Register (see Table 3-16 on page 3-14) and informs its correspondent receivers on the other side through the sideband signals to enter the same power state.

During reset SerDes is held in a power down state with all lanes disabled. After exiting reset, the HyperLink module sends a message via the sideband bus to the remote device requesting its abilities. After the abilities are received, the HyperLink automatically enters an operable state. SerDes is only brought out of reset if either the zero lane bit of the PWRMGT register has been cleared or a transaction is pending. The HyperLink module automatically changes the power mode based on the PWRMGT register settings and the outbound load. By default, the HyperLink leaves the Tx link idle until a transaction is received from its VBUS slave port. Then, the HyperLink enters one lane mode to serve this transaction and the lane power up procedure for one lane is completed. HyperLink dynamically manages its power mode based on the traffic load. When a single lane can not keep up with the traffic load, the HyperLink module automatically enters the four-lane mode, if it is supported. If the load drops to below single-lane performance, the HyperLink module automatically enters the single lane mode by powering down the upper three lanes. If the traffic load is further reduced, the HyperLink automatically enters the zero-lane mode by disabling the single-lane mode and shutting down SerDes until it sees the next transaction. Both transmit and receive are independently controlled so that for certain applications only one direction may be needed.

The transition among different modes is controlled by the lane power management register (Section 3.2.15 "Lane Power Management Control Register (Base Address + 0x44)" on page 3-14).

Figure 2-5 Power State Transition (assuming quadlane, singlelane, and zerolane are all set)



2.6 SerDes Configuration and Clocking

This section describes the SerDes macro provided with the Keystone I HyperLink module.

Note—SerDes module information for KeyStone II devices is not provided in this user guide. Please check for availability of the SerDes User Guide for KeyStone II Devices on the device product page.

The SerDes macro controls the operating rate of the HyperLink, and provides an interface between the HyperLink module transmit and receive interfaces and the external device pins. This section discusses the following topics:

- HyperLink SerDes Operating Rate Configuration
- HyperLink SerDes Receive Interface
- HyperLink SerDes Transmit Interface

Each interface is discussed more detail in the subsequent sections. A set of chip-level registers are provided to access each interface. See the register chapter for more information about these registers. For example code showing how to configure the SerDes, see "HyperLink SerDes Programming Examples" on page A-1.

Note—The SerDes registers are provided at the chip level and are not located in the HyperLink configuration register space. Before accessing these registers, the Kick registers must be programmed to allow access to this register space. See the device-specific data manual for the address of the HyperLink SerDes registers.

2.6.1 HyperLink SerDes Operating Rate Configuration

This section describes how to configure the HyperLink SerDes operating rate. The operating rate of the HyperLink SerDes depends on the configuration of the HyperLink SerDes PLL and the rate scaling factor (RATESCALE) of the receiver and the transmitter. This section also provides information about how to configure the HyperLink SerDes PLL loop bandwidth.

The main purpose of the HyperLink SerDes PLL is to generate a high frequency output clock from a low frequency reference clock (REFCLK). The PLL output frequency depends on the MPY field in HYPERLINK_SERDES_CFGPLL chip-level register, and is calculated by the following equation:

PLL_OUTPUT=REFCLK*MPY

The MPY field can be programmed to several values; however, the PLL_OUTPUT value must be in the 1.5625 GHz to 3.125 GHz range.

Because the PLL uses the low frequency REFCLK input to generate the high frequency output clock, any jitter in the REFCLK can be amplified in the PLL_OUT. If there is too much jitter, it can affect the ability of the transmitter and the receiver to work correctly. To help minimize the effect of the jitter, the LOOP_BANDWIDTH setting is provided. The LOOP_BANDWIDTH field has four settings; a setting that provides a PLL bandwidth in the 8-30 MHz range must be chosen. The PLL bandwidth is defined by the following equation:

PLL_BANDWIDTH = REFCLK/BWSCALE

Table 2-4 shows the BWSCALE values that correspond to different combinations of PLL_OUTPUT frequency and LOOP_BANDWIDTH. Medium bandwidth typically provides the best results, so begin by calculating the PLL_BANDWIDTH using the medium bandwidth BWSCALE for your PLL_OUTPUT frequency. If the PLL_BANDWIDTH for the medium bandwidth configuration is in the 8-30 MHz range, then use the medium bandwidth setting. If the medium bandwidth setting provides a PLL_BANDWIDTH outside the 8-30 MHz range, then repeat the PLL_BANDWIDTH calculation with other LOOP_BANDWIDTH settings to find one that yields a result in the 8-30 MHz range.

Table 2-4 PLL Loop Bandwidth Selection

PLL_OUTPUT Frequency (GHz)	LOOP_BANDWIDTH	BWSCALE	PLL_BANDWIDTH
3.125	Medium Bandwidth	13	PLL_BANDWIDTH=REFCLK/13
	Low Bandwidth	21	PLL_BANDWIDTH=REFCLK/21
	Ultra High Bandwidth	7	PLL_BANDWIDTH=REFCLK/7
	High Bandwidth	10	PLL_BANDWIDTH=REFCLK/10
2.1	Medium Bandwidth	14	PLL_BANDWIDTH=REFCLK/14
	Low Bandwidth	8	PLL_BANDWIDTH=REFCLK/8
	Ultra High Bandwidth	23	PLL_BANDWIDTH=REFCLK/23
	High Bandwidth	11	PLL_BANDWIDTH=REFCLK/11
1.5625	Medium Bandwidth	16	PLL_BANDWIDTH=REFCLK/16
	Low Bandwidth	8	PLL_BANDWIDTH=REFCLK/8
	Ultra High Bandwidth	30	PLL_BANDWIDTH=REFCLK/31
	High Bandwidth	14	PLL_BANDWIDTH=REFCLK/14
End of Table 2-4			

The RATE field in the receive and transmit interfaces provides an additional level of scaling beyond what is provided by the PLL (see the HYPERLINK_SERDES_CFGRX and HYPERLINK_SERDES_CFGTX registers for more information). The RATE field allows 4 different rates to be selected, which can increase or decrease the operating rate of the HyperLink. The transmit and receive rate must be consistent between the transmitter and receiver (e.g. they must be operating at the same rate). The effect of the rate scale factors are shown in table Table 2-5.

Table 2-5 Effect of the RATE Bits

Rate	RATESCALE	Description
Full Rate	0.25	Four data samples taken per PLL output clock cycle
Half Rate	0.5	Two data sample taken per PLL output clock cycle
Quarter Rate	1	One data sample taken per PLL output clock cycle
Eighth Rate	2	One data sample taken every two PLL output clock cycles

Using the below equation, the operating rate, or LINERATE, of the HyperLink SerDes can be calculated using the REFCLK, the MPY factor, and the RATESCALE.

LINERATE=REFCLK*MPY/RATESCALE

Using the maximum frequency of REFCLK*MPY=3.125, and the maximum RATESCALE (full rate), the maximum LINERATE of the HyperLink module is:

LINERATE=3.125/0.25= 12.5 GHz

Table 2-6 shows several example operating rates for different combinations of REFCLK, MPY, and RATE values. Please see the device specific data manual for the REFCLK frequencies supported by the HyperLink module.

Table 2-6 Example Operating Rate Configurations

		LINERATE=REFCLK*MPY/RATESCALE (GHz)			
RefClk (MHz)	MPY	Eighth (11b) RATESCALE=2	Quarter (10b) RATESCALE=1	Half (01b) RATESCALE=0.5	Full (00b) RATESCALE=0.25
156.25	16 (01000000b)	1.25	x	x	x
	10 (00101000b)	x	x	3.125	6.25
	20 (01010000b)	x	x	x	12.5
250	10 (00101000b)	1.25	x	x	x
	12.5 (00110010b)	x	3.125	6.25	12.5
312.5	8 (00100000b)	1.25	x	x	x
	5 (00010100b)	x	x	3.125	6.25
	10 (00101000b)	x	x	x	12.5

The PLL is enabled automatically by the HyperLink module, based on the value of the RESET bit in the CONTROL register. When enabled, the user must poll the LOCK bit in the HYPERLINK_SERDES_STS to ensure that the PLL is operating properly.

2.6.2 HyperLink SerDes Receive Interface

This section discusses the HyperLink SerDes receive interface. The receive interface consists of four data lanes, each of which is configured through the HYPERLINK_SERDES_CFGRX[3-0] chip level registers. The ENRX field in these registers provide the status of the receive modules. The receive modules are enabled automatically by the HyperLink module.

The main task of the receiver is to perform data/clock recovery on the receive signals. The clock recovery algorithms are controlled by the CDR field in the HYPERLINK_SERDES_CFGRXn register. The clock data recover algorithms operate to adjust the clocks used to sample RXp and RXn so that the data samples are taken midway between data transitions. Both first and second order algorithms are provided, and both algorithms use the same basic technique for determining whether the sampling clock is ideally placed, or if it needs to be moved earlier or later.

The first order algorithm operates by making a single phase adjustment each time a threshold is equalled or exceeded. The second order algorithm acts repeatedly according to the net difference between the number of times the selected first order threshold is equalled or exceeded, thereby adjusting for the rate of change of phase.

For each algorithm, there are several phase tracking rates that can be configured to adjust the tracking rate that is used for that algorithm. Although higher order algorithms and higher tracking rates provide better precision, they are cause higher power consumption. If power consumption is a concern, then selecting a lower order algorithm or a lower tracking rate can save power at the expense of phase tracking. The algorithm and tracking rate that provides the best results for phase tracking and power consumption must be evaluated for each system.

The receiver also supports the ability to detect a loss of signal condition, also known as electrical idle. Loss of signal detection can be enabled through the LOS field in the HYPERLINK_SERDES_CFGRXn register. When loss of signal detection is enabled, the LOSDTCT bit in the HYPERLINK_SERDES_STS register can be used to monitor whether a loss of signal condition has occurred.

2.6.3 HyperLink SerDes Transmit Interface

This section discusses the HyperLink SerDes transmit interface. The transmit interface consists of 4 data lanes, each of which is configured through the HYPERLINK_SERDES_CFGTX[3-0] chip level registers. The ENTX field in these registers provide the status of the transmit modules. The transmit modules are enabled automatically by the HyperLink module.

The transmit interface sets up the transmit lanes, and provides options for signal conditioning on the transmit signals. Please see the SWING, TWPRE, and TWPST1 fields in HYPERLINK_SERDES_CFGTXn registers for more information about the configuration options for the transmitter.

2.6.4 HyperLink SerDes Training Process

The Hyperlink peripheral is designed to initialize itself and its remote link partner with minimal CPU intervention. The Hyperlink peripheral will start its side-band signal and SERDES training sequence as soon as it detects the SERDES PLL has achieved lock status (HYPLINK_SERDES_STS[0] = 1).

The PLL enable signal of the Hyperlink SERDES module is tied to the Hyperlink module reset signal. Because of this, when the Hyperlink peripheral comes out of reset the PLL enable bit in HYPLINK_SERDES_CFGPLL register is set internally.

Because the PLL enable bit is set as reset is de-asserted, when a non-zero MPY value is written to the HYPLINK_SERDES_CFGPLL register the SERDES PLL will lock and the Hyperlink training sequence will be triggered.

To ensure that the Hyperlink configuration registers are properly utilized during the initial training sequence, and to ensure robust communication both during the training sequence and in subsequence data traffic, the following sequence of steps should be followed for Hyperlink peripheral initialization.

1. The Hyperlink peripheral is released from reset after normal reset sequence is completed
2. System should assure that both Hyperlink link devices are out of reset and running application code
3. Application should then configure any general Hyperlink registers and Configure the SerDes transmitter and receiver configuration registers
4. The application should then program the PLL configuration register with the proper MPY settings. If the settings are valid, this will result in the PLL locking to the given reference clock at the designated output frequency setting
5. Hyperlink will wait for SerDes PLL lock and then start the Hyperlink side-band signal and SerDes training sequence

In addition to the proper initialization sequence, there are initial analog effects of the SerDes transmitters and receivers that must be accounted for when the Hyperlink SerDes training sequence turns the SerDes transmitters and receivers on for the first time after a power-on-reset state or when coming out of a Hyperlink sleep mode.

The Hyperlink SerDes Control and Status 1 Register (SDCS1) at Hyperlink memory map offset 0x70 contains two bit fields that allow for a controlled number of initial SERDES symbols to be masked by the SerDes receivers during initial SerDes power up or when coming out of a Hyperlink sleep state.

SDCS1[sleep_cnt] control the amount of SerDes symbols that are masked for lanes that enter a sleep/enable state. This allows the internal SerDes power supplies to stabilize before the link is established.

SDCS1[disable_cnt] controls the amount of SerDes symbols that are masked for a SERDES lane that enters a disabled state. This allows the SerDes clock data recovery (CDR) and equalizer to stabilize before the link is established.

Sleep masked symbol count = sleep_cnt x 16 x Unit Interval

Disable masked symbol count = disable_cnt x 16 x Unit Interval

Unit Interval is the bit-period width of the SerDes symbol as determined by the PLL and transmitter settings.

The recommended value for SDCS1[sleep_cnt] is 0xFF. Likewise the recommended value for SDCS1[disable_cnt] is 0xFF. This setting will provide the SerDes transmitters and receivers the maximum amount of time to stabilize after initial power on and after exiting a sleep state.

2.7 HyperLink Protocol

HyperLink provides three types of transactions: read, write, and interrupt events. The following section describes the data flow for read and write operations. The interrupt event is carried by a special packet format. See Section 2.7.5 "Interrupt" on page 2-24 for more information about the interrupt capability of HyperLink.

2.7.1 HyperLink Write Operations

This section discusses the HyperLink write operations. The write process is outlined in Procedure 2-1 and shown in Figure 2-6.

Procedure 2-1 HyperLink Write Operation

Step - Action

1 HyperLink receives a write transaction from its slave VBUSM port, and the write command is written to Outbound Command FIFO.

2 Data is subsequently read from the FIFO and encapsulated in a Write Request packet.

3 The egress address transaction logic overlay controls information into address field.

4 The packet is encoded and serialized before being transmitted to the remote device.

5 The remote device receives the packet, deserializes, and decodes the receive data.

6 The remote HyperLink module stores the received write packet into the Inbound Command FIFO.

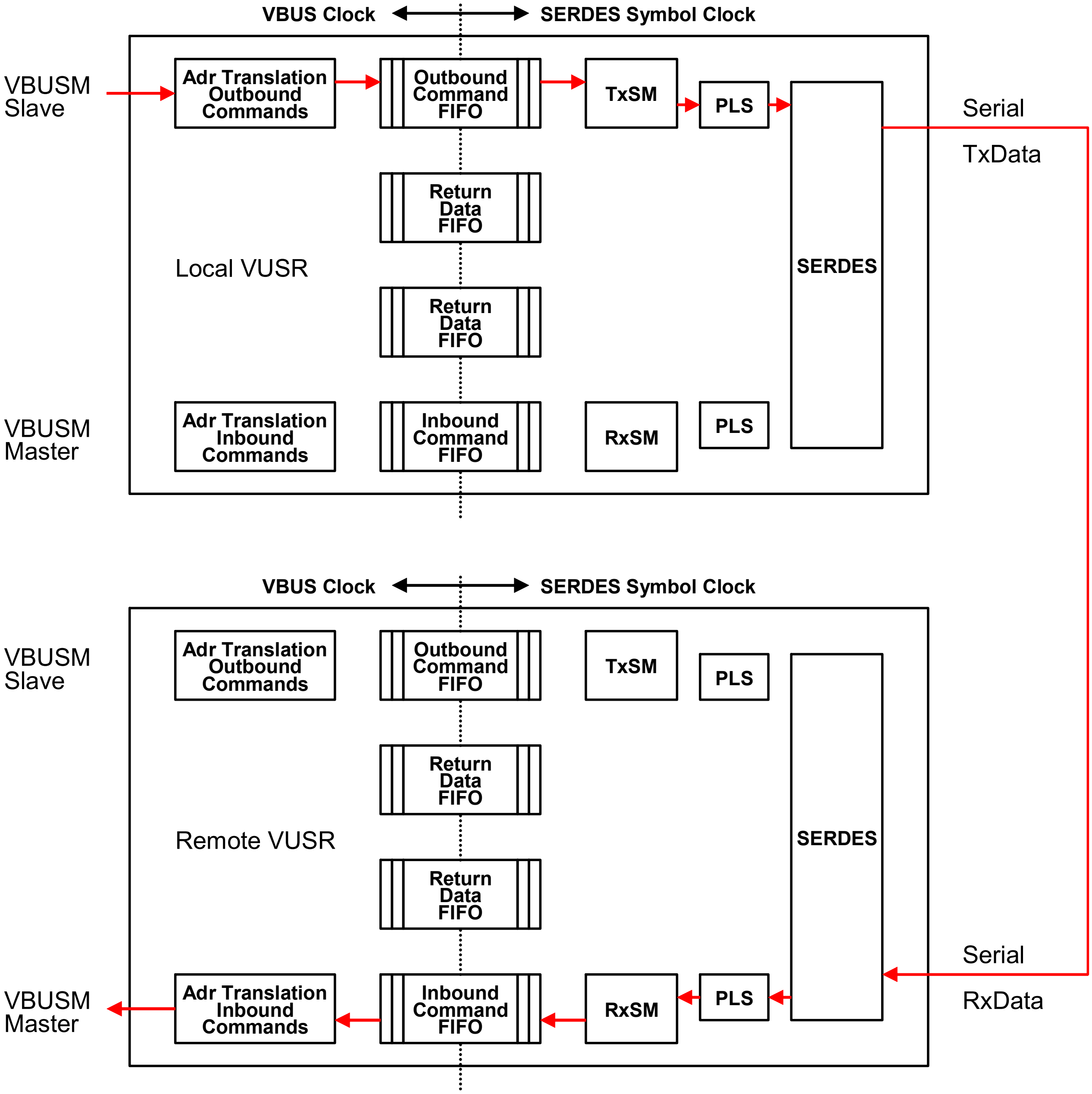
7 The ingress address translation logic generates the new memory-mapped address and other control information such as security and PRIVID for the write transaction.

8 The HyperLink in the remote device initiates a VBUSM master write operation based on the new address and other control information generated by the ingress address transaction

9 The write data lands at the remote device.

10 For a burst write larger than 256B, the write operation is divided into multiple bursts with the maximum burst size of 256B at the remote device. Step 2 to step 9 are repeated for each burst of write data.

Figure 2-6 HyperLink Write Operations



2.7.2 HyperLink Read Operations

This section discusses the HyperLink read operations. The read process is outlined in Procedure 2-2 and shown in Figure 2-7.

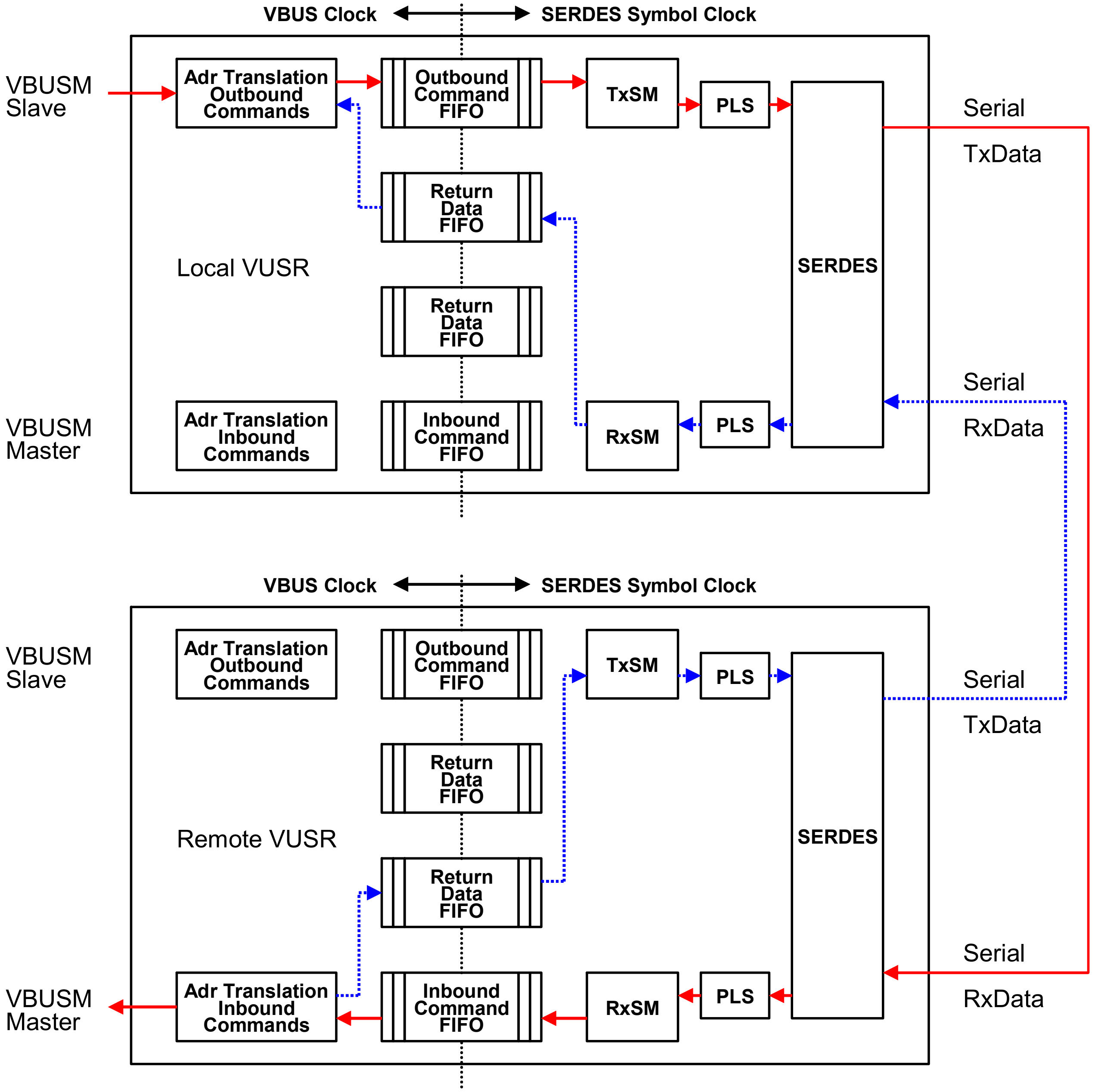
Procedure 2-2 HyperLink Read Operations

Step	Action
1	HyperLink receives a read request from its VBUSM slave port.
2	The read request is stored in the Outbound Command FIFO.
3	The read command is read from the Outbound Command FIFO and the egress address translation logic modify its address field and overlay the control information before it is encapsulated in a Read Request Packet.
4	The packet is encoded, serialized, and sent to the remote device.
5	The remote device receives the Read Request Packet through its HyperLink interface.
6	The received packet is deserialized, decoded, and written to the Inbound Command FIFO.
7	The read request packet is read from the Inbound Command FIFO and initiates a read request based on the address and control information which the ingress address translation logic generates from the received address field.
8	The read request is sent out through the HyperLink master VBUSM port to one of the slave end point in the remote device.
9	When the HyperLink in the remote device receives the read return data, the data is written to its Return Data FIFO before being serialized.
10	The read return data is sent out by the HyperLink interface in the remote device.
11	The read return data packet is received by the HyperLink interface, and it is Step 12:The read data is transferred on the local VBUSM slave interface.
12	The read data is transferred on the local VBUSM slave interface.

Steps 9 to 12 are repeated for all data phases within the read burst. HyperLink does not wait for the completion of the read data before it sends out the read return data packet. It forms the read data packet on every read return burst. The read request is larger than 256B, it is divided into multiple read requests, each of which has maximum 256B burst.

The data flow between two connected HyperLink modules is shown with arrows in Figure 2-7 on page 2-15, with the read access originating from the Local HyperLink device. The read data is returned by the remote HyperLink device and displayed with the dotted arrows.

Figure 2-7 HyperLink Read Operations



2.7.3 HyperLink Format

HyperLink operates on 64-bit quanta. Each word in HyperLink is 64 bits and it is sent as a single element. For each transaction, HyperLink transmits to the remote device through control words and data words.

Each HyperLink packet includes one or two control words and multiple data words. The Control Word and Control Word Extension bit definitions are defined in section 0 and 0 respectively. The Control Extension field is only included in the packet if the packet type indicates extended control (ext=1). This provides a mechanism to augment commands sent in the Control Word. The Control Extension is used to deal with configuration packet types, which is used to access the registers in the remote HyperLink module. The HyperLink module in KeyStone devices does not support control extension.

HyperLink supports the following packet formats and how it is constructed:

- Read request: c. only one control word
- Write request: cD or cdd..dD
- Interrupt packet:c
- Read data return: cdddD
- Write request with byte enable: cdd..dmD. An example of cddmD means packet wrote 32 bytes where the bytes 16-23 contained disabled byte enables

Where:

I	Idle
I#	Packet continuation where the # is the 'c' from the packet continued.
c	Control Word: Command, length, address, TransactionID, etc.
e	Extension Control Word:
d	data
D	the last data word
m	data with mask bytes

A data burst always starts with a control word. The control word defines the characteristics of the data burst. When the packet is throttled by the remote HyperLink receiver, the idle code is inserted into the data stream. When the flow control is lifted, the transaction can be continued with a control word. For example, a write burst is flow controlled: cdddIIIcddddD.

The example above shows what can happen to a Write Burst due to remote FIFO state changes. In HyperLink, the outbound transactions and the return data share the same physical link to the remote device. The outbound transactions and data return can be interleaved. However, the data return has higher priority than the outbound transactions.

This is an example of an outbound write transaction interrupted by a read return data burst. The 1 indicates the control word for the data return channel and 0 indicates the control word for the outbound write transaction:

IIIIcddddIcdddIII1ddddII0dddddddddddddIIIII0ddDIIIII1DIIII

From the idle stream, HyperLink receives a command, length, address, and start receiving data. A flow-enable was received for the command channel, but there is data to return so the Flowed is followed by a channel 1 descriptor (the command for return data actually indicates a channel 1), and the channel 1 packet is now under way. A flow is then received for channel 1, but it is soon disabled so the channel 1 packet continues. Again the flow is enabled for channel one, quickly after flow is released for channel 0 so the data continues for channel 0 when a flow is received again for channel 0. Channel 0 then receives a flow disable, completed its packet, and followed by channel 1 flow disable where the channel one packet is also completed.

Whenever a channel is resumed—either from idle or another channel—the control word is resent. Resending the control allows the receiver to identify where to associate the data burst.

2.7.3.1 General Bus Control Word

The General Bus Control Word transfers bus commands that provide the capability to read and write remote HyperLink devices. Currently, there are four types of control words supported: write post, read, interrupt and read data return.

The control word for read and write share the same format.

Figure 2-8 Control Word for Read and Write

Bit 63 to 58	Bit 57	Bit 56 to 54	Bit 53 to 52	Bit 51 to 48	Bit 47	Bit 46 to 44	Bit 43	Bit 42 to 35	Bit 34 to 7	Bit 6	Bit 5	Bit 4 To 0
Reserved (all zeros)	dbg	cclsize	Camode	extAdr	0	CMD	cfg	length	Address	Priv	Pri	Order ID

Figure 2-9 Control Word for Read Return

Bit 63 to 48	Bit 47	Bit 46 to 44	Bit 43	Bit 42 to 35	Bit 34 to 30	Bit 29 to 20	Bit 19 to 17	Bit 16 To 11	Bit 10 8	Bit 7 To 0
Reserved (all zeros)	0	CMD= 0b111	cfg	Length	Align	Reserved	Status	Priv	Pri	Order ID

The control word for an interrupt packet is different from the read/write control word and contains some interrupt-specific information.

Table 2-7 Control Word Encoding (Part 1 of 2)

Field	Number of Bits	Description
DBG	1	EMUDBG - Indicate that a transaction comes from the debug subsystem. This comes from the VBUS transaction sideband information. 0b1:transaction comes from the debug subsystem 0b0: non-debug transaction
CMD	3	Command Type. Indicates the type of bus command being serviced. 000b = write 100b = read 110b = write interrupt 111b = return data/status All other values are reserved.
CCLSIZE	3	Cache line wrap size. This field indicates the cache line size the cache line wrap operates on. This field comes from the received VBUS transaction.
CAMODE	2	Address Mode. This field indicates the Address mode of the command. This comes from the received VBUS transaction.
EXTADR	4	4MSB bit of the address
ADDRESS28	28	28 LSB bits of the address indicates the virtual address the data is to be written or read.
CFG/EXT	1	Configuration/Extension Flag. This flag indicates that a configuration access or the extension control word is present after the control word. It is set to 0
LENGTH	8	Length. Indicates the length in bytes of the transaction.
PRIV	1	Indicates the privilege that the command has been sent with. This comes from the VBUS transaction side band signals directly. 1 = supervisor mode 0 = use mode

Table 2-7 Control Word Encoding (Part 2 of 2)

Field	Number of Bits	Description
PRI	1	PRI indicates the priority of the request. HyperLink only supports the two priorities for outbound transactions. If the received command's priority is between 0 and 3, the PRI is set to 0. If the received command's priority is between 4 to 7, the PRI is set to 1. 0 = priority 0 1 = priority 4
ORDERID	5	Order ID. Indicates the transaction identifier associated with the request. All request with the same ORDERID will be services in the same order as the command arrival. Commands with differing ORDERID fields can be processed in any order. The ORDERID field is assigned by hardware and it is transparent to the user.
STATUS	3	Status. This field is the status for read and write responses.

Figure 2-10 Control Word for Interrupt Packet

Bit 63 to 48	Bit 47	Bit 46 to 44	Bit 43	Bit 35 to 32	Bit 31 to 24	Bit 16 To 11	Bit 10 8	Bit 7 To 0
Reserved (all zeros)	0	CMD= 0b110	cfg	MPU select	vector	Priv	Pri	Order ID

Although multiple bits are assigned for the PRIV and PRI fields, only the LSB bit is used.

The MPU select field carries the information for microprocessor select bit. And the vector field is assigned 8 bits, but only the last 5 LSB bits are used in KeyStone devices. The unused bits are set to zeros.

Even though eight bits are assigned for order ID, only the five LSB are used. The three MSB bits are zeros.

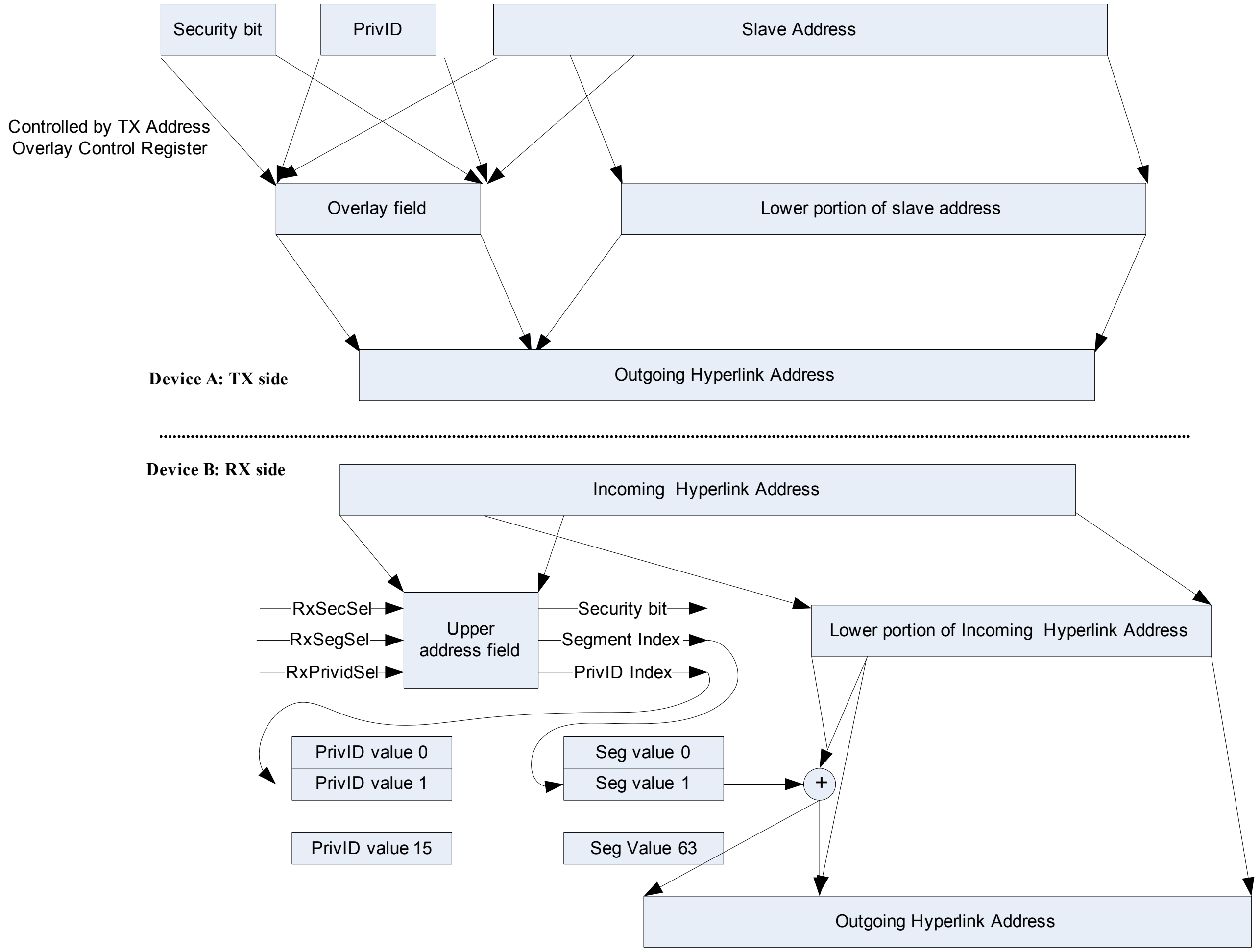
2.7.4 Address Translation

HyperLink has independent ingress address translation blocks and egress address translation blocks. The main function of the ingress address translation is to overlay the control information into the address field. The main function of the egress address translation is to remap the incoming address to different memory regions. HyperLink supports up to 64 different memory regions. The starting address of each memory region can be placed on any 64 KB address boundary, and the size of each memory mapped region can be sized to a power of two, starting at 256 B. In KeyStone devices, the maximum memory region size is 256MB. The address translation is done on a per-packet basis.

Some of the transaction characteristics have been embedded into the control word, such as priority, supervisor, and use mode, etc. The PRIVID and security characters are embedded in the address field.

Figure 2-11 shows how the address translation done at both the TX and RX side. On the TX side, the TX overlay control register determines how the security bit and PRIVID overlay on top of the slave address to form the outgoing address.

Figure 2-11 Address Translation in HyperLink

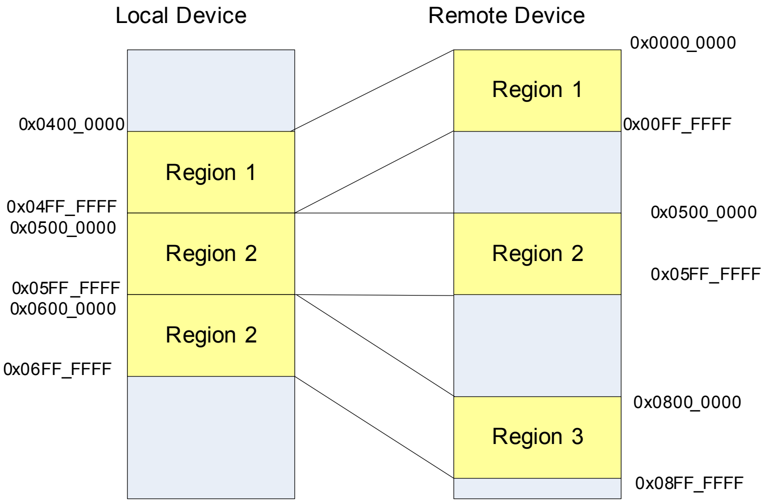


On the RX side, the address translation logic regenerates the address, security and the PrivID field for the received transactions. The RX address selector control register determines which portion of the incoming address to use to regenerate those fields. The RXSECSEL field determines which bit of the address is used to determine how to choose the security bit between RXSECHI and RXSECLO. The security selection can be disabled by setting RXSECHI and RXSECLO to the same desirable value. On the RX side, there is a PRIVID table with 16 entries; each entry can be set individually to different values. The possible PRIVID value is 0 to 15. The RxPrividSel field determines which portion of the incoming address determines the index to the PrivID table. On the RX side, the address can be mapped to 64 segments. Each segment has its own register to define the starting address and size of the segment. The RxSegSel field is used to determine which of the 64 segments the current incoming address is mapped to. The new outgoing address is formed based on the segment starting address, segment size, and lower portion of the incoming address.

An added feature of memory management is that the PrivID and secure bits from the slave interface can be used as selection criteria.

Figure 2-12 shows one possible memory map.

Figure 2-12 Memory Map Example



2.7.4.1 Address Translation on the TX Side

The egress address translation is controlled by the TX Address Overlay Control register (Figure 3-8). It determines how the address is trimmed and overlays the control information such as security and PrivID for the remote device.

Tx address generation is divided into three steps. First, a portion of the ingress address is selected as the TX address controlled by the TxigMask field. Then the PrivID of the incoming transaction is overlaid, which is controlled by the TxPrividOvl field. Finally, the security bit of the incoming transaction is overlaid to form the final Tx address to send to the remote device. Both PrivID and the security bit have 16 different positions to overlay on the Tx address. When there is a conflict for the overlay position, it is determined by the following priority order (from high to low):

- Security
- PrivID
- Address

2.7.4.2 Address Translation on the RX Side

Ingress address translation is more complex than egress address translation. Different portions of the address can be selected to remap the address, PRIVID, and security fields.

The RX Address Selector Control Register (Figure 3-9) configures which Rx Address bits are used to select the Secure, PrivID, and Segment/Length value arrays. This register also holds the secure value when the secure selection is one or zero.

2.7.4.2.1 PrivID Mapping on the RX Side

On the RX side, HyperLink has a PrivID table with 16 entries. Each entry can be independently program a 4 bits PrivID value. Based on RxPrividSel field from Rx Address Selector Control Register, portion of the ingress address is used as the index to access this PrivID table to determine the PrivID value for the incoming packet.

The registers used to program and check the content of the RX PrivID remapping table are the Rx Address PrivID Index Register (Table 3-11) and the Rx Address PrivID Value Register (Table 3-12). The Rx Address PrivID Index Register is used to select which PrivID array element is written or read from the Rx Address PrivID Value Register.

The following steps are used to program the PrivID table:

- Write to the RX Address PrivID Index Register. The four LSB bits are the index of the entry to the program. The 28 MSB bits are all zeros.
- Write to the RX Address PrivID Index Value Register. The four LSB bits are the PrivID values stored in entry specified in the first step. The 28 MSB bits are all zeros.

The steps to check the PrivID table content are:

- Write to the RX Address PrivID Index Register. The four LSB bits are the index of the entry to the program. The 28 MSB bits are all zeros.
- Read the RX Address PrivID Index Value Register. The four LSB bits of the read return data is the PrivID value stored in the entry specified by the first step.

2.7.4.2.2 Address Remapping on the RX Side

The address remapped at the RX side is determined by the segment address and offset mask. Based on RxSegSel field in the RX Address Selector Control register, the HyperLink uses certain fields of the incoming address as an index to access the segment/length table to determine the segment address. The RxSegSel field is also used to determine what the OFFSET_MASK value is for the address remapping at the RX side. There are 16 offset masks in HyperLink. The segment/length table has 64 entries; each entry can be programmed independently through the Rx Address Segment Index Register (Figure 3-12) and RX Address Segment Value Register (Table 3-14). The Rx Address Segment Index Register is used to select which Segment/Length array element is read or written using the Rx Address Segment Value Register. The Rx Address Segment Value Register is used to read or write the selected Segment/Length array element.

The translated Address at the RX side is equal to the Segment Address + (Rx Address & OFFSET_MASK). The segment Address[31:16] comes from the RXSEG_VAL from the segment/length table, and the segment Address[15:0] = 0x0000.

Each segment can be configured to have its own segment address and segment size. The segment size can range from 512 B to 256 MB. Each segment should be started at the 64 KB address boundary.

2.7.4.2.3 Example of Address Translation

This example is based on the TX and RX address registers having the following configuration:

Table 2-8 Example of Address Translation

Register Name	Local HyperLink Module Setting	Remote HyperLink Module Setting
TX ADDRESS REGISTER		
TXIGMASK	8 (means the address mask is 0x01FF_FFFF). Don't care.	Don't care
TXPRIVIDOVL	10 (means that PRIVID is overlaid at address bit 29:26)	Don't care
TXSECOVL	9 (means that security bit is overlaid at address bit 25)	Don't care
RX ADDRESS REGISTER		
Segment Selector	Don't care	8 (the address mask is 0x00ffffff and use address bit 29:24 for segment/mask selection)
PRIVID Selector	Don't care	10 (Using address bit 29:26)
Secure Selector	Don't care	9 (using address bit 25)
Segment for Seg 0x1	Don't care	0x1234
Length for Seg 0	Don't care	9
PRIVID for PRIV Index 7	Don't care	3
RXSECHI/RXSECLO	Don't care	0

If a transaction is received by HyperLink with address 0x0408_0050, with PRIVID=7 and Security signal =1, the following actions are carried out:

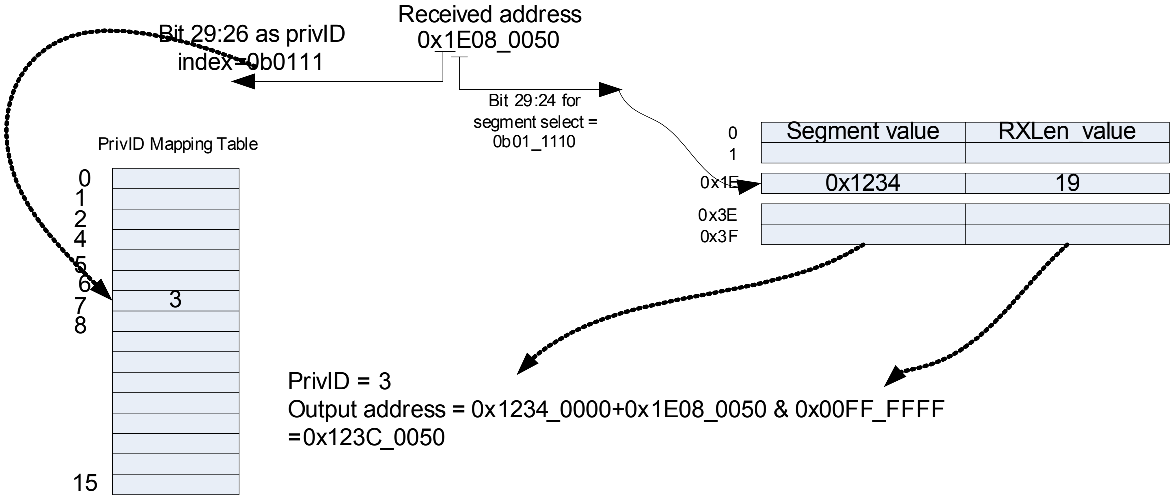
After applying the TX address mask: Because Txigmask=8, the address mask is 0x01FF_FFFF. Therefore, the lower portion of the address will be 0x0408_0050 & 0x01FF_FFFF= 0x0008_0050.

Overlay the PrivID value =7 at address bit 29:26, it gets 0x1C08_0050.

Overlay the security bit at address bit 25, it gets 0x1E08_0050.

Therefore, the address sent to the remote device through the HyperLink is 0x1E08_0050.

Figure 2-13 Address Translation at RX Side



On the remote side, HyperLink receives this transaction with address 0x1E08_0050. The segment selector is based on address bit 29:24, which is 0x1E for address 0x1E08_0050. The address mask is 0x00ffffff. So the lower portion of the RX address is 0x1E08_0050&0x00FF_FFFF=0x0008_0050. For segment 0x1E, the segment address is 0x1234 and the rxlen_val=19. This means that this segment size is 1MB (0x0010_0000).

Therefore, the translated address on the RX side is 0x0008_0050 + 0x1234_0000 = 0x123C_0050.

For PrivID, because the Rx Address PrivID Selector =10, it means to use the incoming address bit 29:26 to select PrivID array. Bit 29:26 of the incoming address 0x1E08_0050 is 7. The value of PrivID array 7 is 3. Therefore, the outgoing transaction PRIVID value is 3.

For the security signal, where the Rx Address Secure Selector=9, address bit 25 is used for the security selector. Bit 25 of the incoming address 0x1E08_0050 is 0b1. Therefore, the outgoing security signal uses the RXSECHI field in the Rx Address Selector Control Register. In this case, the RXSECHI field is 1.

Therefore, in this example, the incoming address 0x1E08_0050 is translated to address 0x123C_0050 with PRIVID=3 and the security signal set to high.

Transactions that cross map region boundaries may violate CBA slave endpoint access rules. The behavior of such a transaction is undefined. The transaction must be limited to one region for correct operation.

Any transaction that violates the Segment Length value will have the address[31:5] zeroed and the byte enables zeroed as to prevent any write or read side effects from occurring.

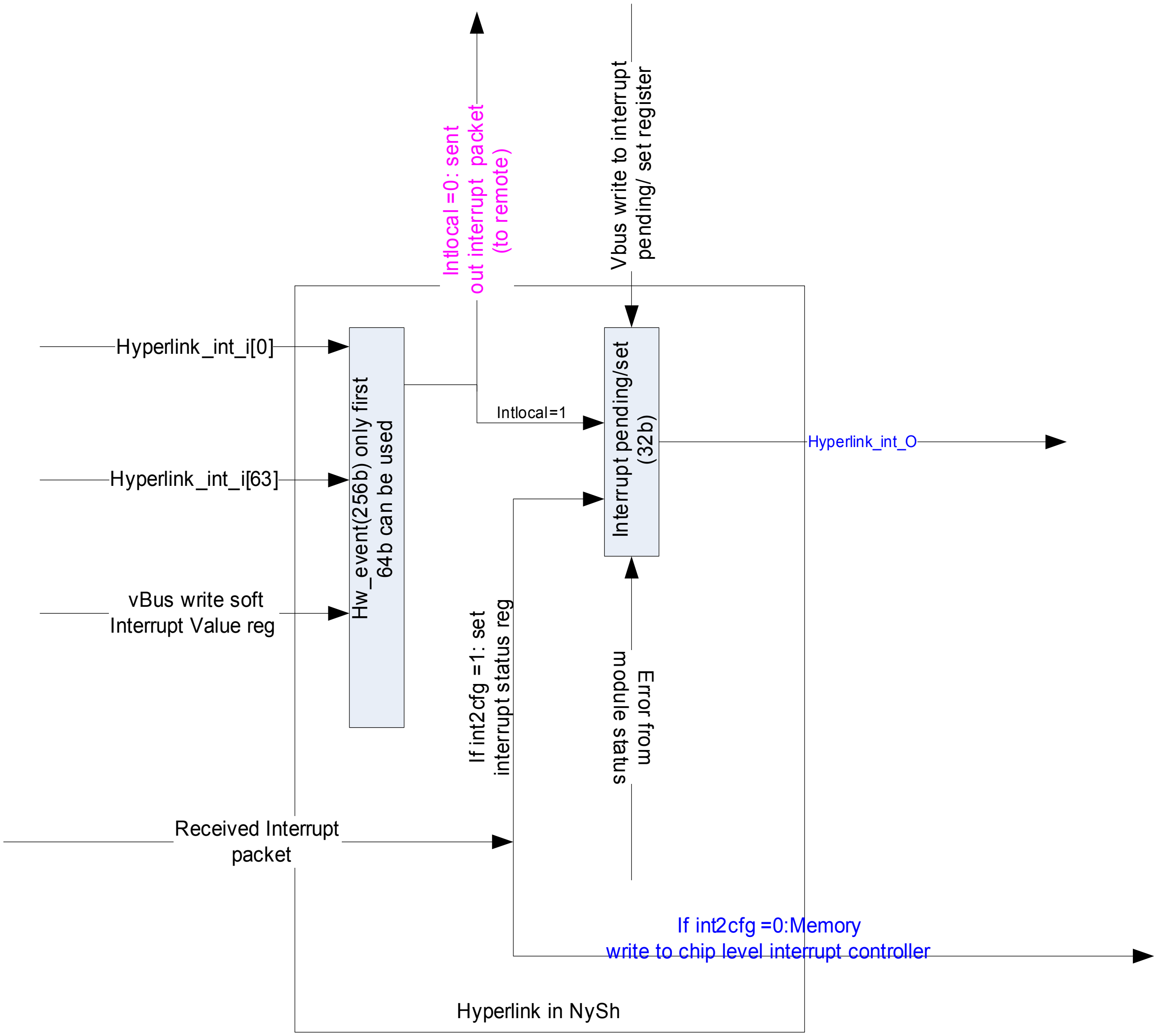
2.7.4.3 CBA Burst Split Operations

To ensure quality of service, HyperLink segments the large transactions into bursts of up to 256 bytes. The largest transport packet can support 256 bytes of data. In KeyStone devices, both read and write bursts are limited to 64 B and they are split at the 64 B address boundary.

2.7.5 Interrupt

Each HyperLink module is designed to provide maximum flexibility for detection, forwarding, mapping, and generation of interrupt information.

Figure 2-14 Interrupt Architecture in HyperLink


Intlocal =0: sent out interrupt packet (to remote)
Vbus write to interrupt pending/ set register
Hyperlink_int_i[0]
Hyperlink_int_i[63]
vBus write soft Interrupt Value reg
Hw_event(256b) only first 64b can be used
Intlocal=1
Interrupt pending/set (32b)
Hyperlink_int_O
If int2cfg =1: set interrupt status reg
Error from module status
Received Interrupt packet
If int2cfg =0:Memory write to chip level interrupt controller
Hyperlink in NySh

Detection refers to the capturing of either hardware interrupts via the HYPERLINK_INT_I[x] pin or writes to the SW_INT register into the HW_PENDING event holding latch. Forwarding refers to the process that removes events from the HW_PENDING register and sends them to the remote device or adds them to the local interrupt pending/set register based on the value of the INTLOCAL bit. Mapping refers to the process of receiving an interrupt packet from the remote and forwarding it to a device interrupt controller or to the local interrupt pending/set register via the int2cfg configuration bit. Generation refers to the process that drives the interrupt output pin from the contents of the interrupt pending/set register. The interrupt pending/set register can be set locally, by a remote configuration register write, or the reception of an interrupt packet destined for the configuration space via the INT2CFG bit.

2.7.5.1 Interrupt Detection

Interrupt detection captures the hardware or software event into an internal HW_EVENT[x] holding latch when enabled. HyperLink can hold 64 events. Each event is configured individually through a set of registers: the interrupt control index register and interrupt control value register. The interrupt control index register, Table 3-21, is used to control which event the internal control register is read or written via the Interrupt Control Value register. The interrupt control value register, Figure 3-21, defines the characteristics of the correspondent event. In this version of HyperLink, because there are only 64 events defined, only the six LSB bits of the ICIDX field are used and the two MSB bits should be zeros.

Each HW_EVENT must be individually configured. The following steps show how to configure HW_EVENT[x], while x is between 0 to 63:

- Write to the Interrupt Control Index register at base address+0x60 with value x (x is a decimal value)
- Write to the Interrupt control value register at base address +0x64 with a desirable value:
  - INTEN field: when set to zero, it means the event coming from the HYPERLINK_INT_IN[x] will not be detected. In another word, the hardware interrupt input is disabled for this HW_EVENT[x];
  - SIEN: when set to zero, it means that software can not trigger HW_EVENT[x].
  - INTTYPE: 0 means level sensitive and 1 means pulse sensitive.
  - INTPOL: set the polarity. When it is set to 1, it indicates active low, it is zero, it indicates active high.
  - ISEC: when it is set to 1, only the secure master can trigger the software event.
  - DNID: current not used and set to zeros.
  - MPS: only eight MicroProcessors are supported, so only the 3LSB bits are used. the MSB bit should be always set to 0. This field is used to send interrupt packet to the remote device.
  - VECTOR: when the INT2CFG is set to 1 in the control register, Table 3-3, this field indicates which bit of the interrupt pending register to set if there is an event trigger at HW_EVENT[x]. When INT2CFG is set to 0 in the control register, if HW_EVENT[x] is trigger, the interrupt packet is sent to the remote device with MPS and vector field. The vector field is either indicates which bit of the remote pending register to set or used as an index to get interrupt address depending on the int2cfg setting at the remote device.

The steps to read the interrupt control value for a specific HW_EVENT[x] are:

- write to Interrupt Control Index register at base address+0x60 with value x (x is a decimal value)
- Read Interrupt control value register at base address +0x64

If x is larger than or equal to 64, writing to the interrupt control value register results in no operation and the read from the interrupt control value registers returns all zeros.

The fields INTEN, INTTYPE, and INTPOL are for hardware events; the fields ISEC and SIEN are for software events.

In KeyStone devices, events 0 to 63 are connected on the chip level. Events 0 to 31 come from the chip level Interrupt controller and events 32 to 64 are from queue-pending signals from the QM to monitor some of the transmission queue status. Because these 64 signals attached to HyperLink are active high, when the INTPOL bit is set to 0b0, a non-empty queue status can trigger an event. When the INTPOL bit is set to 0b1, an empty queue status can trigger an event.

The queue pending signals are level-based signals to indicate whether the transmission queue is empty. The interrupt signals from the chip-level interrupt control are pulses of one CPU/6 clock cycle. Because HyperLink operates at CPU/2 speed, if these interrupt inputs are set to be pulse bases, the HyperLink interprets them as three back-to-back interrupts. Therefore, the interrupt inputs should be also set to level-based. The INTTYPE field for all 64 events should be set to 0b0.

The INTEN field for these 64 events should be set to 1 to allow the capture of the input events. If any of the input events are not desired, the INTEN field should be set to 0b0 to mask the event.

When both INTEN and SIEN are cleared to zero, the HW_EVENT[x] cannot capture any events. When both INTEN and SIEN are set to 1, both HYPERLINK_INT_IN[x] and software can trigger an event. HyperLink has no way to differentiate which source triggers the event. So it is recommended that either INTEN or SIEN be set to 1.

An End-of-Interrupt (EOI) function exists to re-enable the detection of active-level interrupts on the interrupt input pins by writing the Generate Soft Interrupt Value Register. If the system attempts to write to the Generate Soft Interrupt Value Register with a vector of 0xFFFF, any level interrupts still pending will re-event the HW_EVENT register. To re-enable a single hardware interrupt, the software writes to the SW_INT register with a 0xFEYY, where YY is the particular hardware interrupt to be EOI'd. In addition, the first 64 events can also be used as software interrupts if the SIEN bit is set by writing to the Interrupt Control Value register.

The Generate Soft Interrupt Register (Figure 3-7) should be written with a vector of the hardware index of the interrupt that is enabled for software interrupts. If the ISEC bit is also set, the csecure interface pin must be set to enable the software interrupt. This register is also used to EOI HYPERLINK_INT_I hardware interrupts programmed in level mode.

The following example shows the effects of writing to the generate soft interrupt value register. The value below is shown in hex format, and 'x' means 'do not care'. Only the value listed below is valid, and writing to the generate soft interrupt value register with another value has no effect.

Xxxx_FFFF: EOI all the level sensitive interrupt inputs.

Xxxx_FEMN: When 0xMN is between 0x00 and 0x3F, EOI the level interrupt inputs for HW_event[0xMN]. Otherwise, no operations are associated with it.

Xxxx_00MN: When 0xMN is between 0x00 and 0x3F, it triggers a software interrupt by setting HW_event[0xMN] if the SIEN bit of the interrupt control value for HW_event[0xMN] is set to 1.

Note—It is possible to trigger HW_event[0xMN] from both software and hardware inputs. The application needs to determine whether it is a desirable behavior. If it is desirable to avoid the ambiguity of an interrupt source, the interrupt control value for HW_event[0xMN] needs to be configured correctly, so that only software or hardware can trigger an event.

2.7.5.2 Interrupt Processing

HyperLink has a control register to specify how to process the interrupt events, including the events detected in the HW_EVENT registers and the interrupt events sent by the interrupt packets from a remote device. The control register (Table 3-3) also determines other operations of the HyperLink module such as reset and loopback mode.

Interrupt forwarding processes the interrupts in the HW_EVENT[x] based on the INTLOCAL bit setting. If the INTLOCAL bit is cleared, the event is sent to the remote device using an interrupt packet. The interrupt packet contains the vector and the microprocessor select (MPS) fields from interrupt control value register for that HW_EVENT. If the INTLOCAL bit is set, the event is placed into the local INTERRUPT_PENDING_SET register. The bit set in the INTERRUPT_PENDING_SET register is controlled by the vector field of the interrupt control value register.

2.7.5.3 Interrupt Mapping

The int2cfg field in the control register determines how to process the incoming interrupt packets. When int2cfg is clear, the interrupt status is extracted from the received interrupt packet and written to the register indicated by the Interrupt Pointer Register for the particular micro processor select (MPS) value. Currently HyperLink supports eight such interrupt-pointer-register values. This register may be anywhere in memory-mapped space. For example, the Interrupt Pointer Register may contain the memory-mapped location of an interrupt status/set register in the device's interrupt controller. Alternatively, the Interrupt Pointer Registers may point to the memory-mapped location of its own INTERRUPT_PENDING_SET register. The Interrupt pointer Index Register (Table 3-21) and the Interrupt Pointer Value Register (Figure 3-21) are used to configure these eight interrupt pointer register addresses. The 4LSB bits field in the Interrupt pointer index register should be between 0 and 7. Any value outside this range will result in no operation to modify the interrupt pointer value.

When the int2cfg bit is set, the interrupt packet is to used to set the INTERRUPT_PENDING_SET register. The bit position is determined by the vector value carried by the incoming interrupt packets.

2.7.6 Interrupt Generating

Interrupts are generated by setting bits in the interrupt pending/set Register. There are 32 bits in the interrupt pending/set register and each bit can be set independently. When the int2cfg bit is set, the interrupt pending/set register can be set through the incoming interrupt packet by a remote device. Another way to set the interrupt pending/set register is by the local device through a memory write to the interrupt pending/set register. After the bits are set in the interrupt pending/set register, the HyperLink module generates an interrupt output, (Figure 2-15).

The HyperLink has two interrupt signals, one for pulse interrupt output and one for the level-based interrupt output. In KeyStone devices, only the pulse based interrupt signal is hooked up on the chip level, connecting to the chip level INTC, which further connects to GEM and EDMA0. The level-based interrupt output is not used in KeyStone devices.

The interrupt pending/set register can also be cleared by writing to interrupt status/clear register (Table 3-6). The interrupt status/clear register can be used either to clear each bit in the interrupt pending/set register or to clear all the bits.

After the software clears one or multiple bits in the interrupt pending/set register through either interrupt priority vector status/clear register or interrupt status/clear register, if interrupt pending/set register is not equal to 0x0000, a new pulse is generated on the pulse-based interrupt output. When the software clears all the bits in the interrupt pending/set register, the HyperLink de-asserts the level-based interrupt output.

Among all the 32 bits in the interrupt pending/set register, the HyperLink interprets bit 0 as the highest priority while bit 31 is the lowest priority. Reading the Interrupt priority vector status/clear register (Figure 3-4) is used to report the highest interrupt asserted in the interrupt pending/set register. Writing to the Interrupt priority vector status/clear register is used for clearing the individual bit in the interrupt pending/set register.

2.7.6.1 Setting Interrupt Pending Register by Module Status

When the INTENABLE field in the control register (Table 3-3) is set to 1, the module status interrupt is posted to the interrupt pending register. The INTVEC field of the control register indicates the bit position to be set by the module status.

2.7.6.2 Setting Interrupt Pending Register by Writing Interrupt Pending Register

Writing to Interrupt Pending register (Table 3-7) with a non-zero value sets the interrupt pending register. Any non-zero value in the interrupt pending register can trigger HYPERLINK_INT_O.

Writing 0x0000 to the Interrupt pending register will EOI the HYPERLINK_INT_O pin. If the Interrupt pending register is a non-zero value, another interrupt will be triggered at HYPERLINK_INT_O.

2.7.6.3 Setting Interrupt Pending Register by HW_event

When Int2Local bit is set to 1, the event captured in the HW_EVENT is forwarded to the interrupt pending register. For HW_event[x], the vector value in the interrupt control register for HW_EVENT[x] determine which bit in the interrupt pending register is set when HW_event[x] is triggered, either by hardware input or software.

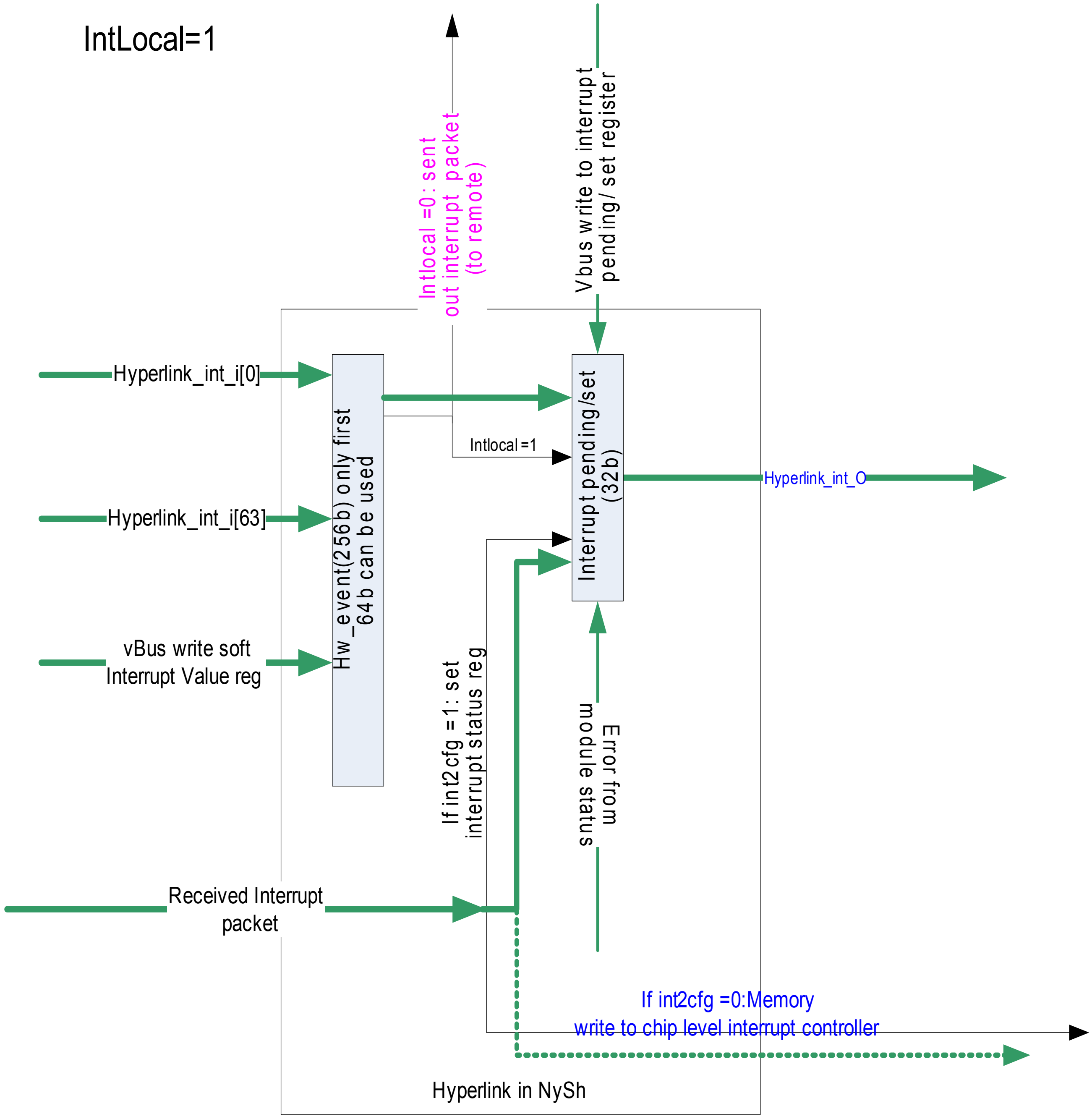
2.7.6.4 Setting Interrupt Pending Register by Remote Device

The Interrupt pending register can also be set by the remote device through the interrupt packet, when the int2cfg is set to 1. The vector value carried in the interrupt paced determines which bit to set in the Interrupt pending register by the received interrupt packet.

2.7.6.5 Clearing the Interrupt Pending Register

Each bit in the Interrupt pending register can be cleared individually. Writing to the Interrupt status/clear register (Table 3-6) can be used to clear one or multiple interrupts. Writing to the priority vector status/clear register clears an individual interrupt.

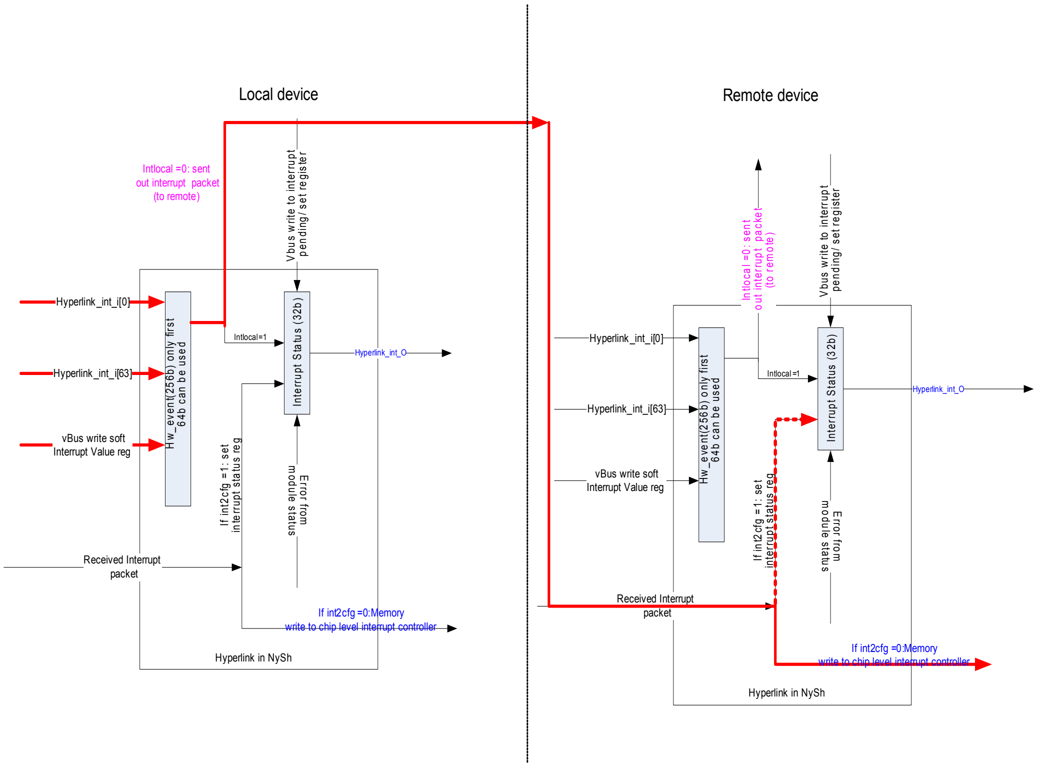
Figure 2-15 Trigger Local Interrupt Hyperlink_int_O


IntLocal=1
Intlocal =0: sent out interrupt packet (to remote)
Vbus write to interrupt pending/set register
Hyperlink_int_i[0]
Hyperlink_int_i[63]
vBus write soft Interrupt Value reg
Hw_event(256b) only first 64b can be used
Intlocal=1
Interrupt pending/set (32b)
Hyperlink_int_O
If int2cfg =1: set interrupt status reg
Error from module status
Received Interrupt packet
If int2cfg =0:Memory write to chip level interrupt controller
Hyperlink in NySh

2.8 Passing an Interrupt to a Remote Device

Figure 2-16 shows how to pass an Interrupt to a remote device through the interrupt packets.

Figure 2-16 Interrupt Remote Device Through HyperLink



For a local device, the INTLOCAL bit must be 0x0 so any event from the HYPERLINK_INT_I pin or software triggered interrupt to HW_EVENT will be transported to the remote device through the interrupt packet. The interrupt packet includes the MPS and vector field for that HW_EVENT.

On the remote side, the interrupt packet is received. If int2cfg for the remote device is set to 0x1, the received interrupt packet will set one of the bits in the interrupt status register. The bit position is determined by the vector value carried in the interrupt. If the int2cfg for the remote device is set to 0x0, HyperLink sets the chip-level interrupt control register. The chip-level interrupt control register address is based on the MPS field carried in the interrupt packet. And the vector field determines which bit of the chip-level interrupt control register is set.

2.9 Reset Considerations

HyperLink supports reset through chip reset and software reset. The reset triggered by the chip reset will reset all logic in HyperLink module. The HyperLink module can also be reset by writing to the control register. Because the HyperLink does not abort transaction in flight if one of the ends of the link is reset, special care should be taken when asserting reset on one of the HyperLink devices after the two HyperLink devices have established the link. To avoid system hang, it is recommended to set the SERIAL_STOP bit to allow all the outstanding transactions to complete before the HyperLink is put into reset. This applies to both device reset and software reset.

2.10 Initialization

TBD

2.11 DMA Event Support

An event from HyperLink does not trigger an EDMA transaction directly. HyperLink has a master port that can issue a transaction on behalf of itself. The EDMA can send transactions to HyperLink the same way it sends transactions to a memory endpoint.

2.12 Power Management

HyperLink provides several low-power states for power saving. Power management is transparent to the user after correctly configuring the power management configuration registers.

2.13 Emulation Considerations

Emulation suspend is not supported in HyperLink.

Chapter 3

Registers

This section describes the registers for HyperLink module and the HyperLink SerDes.

3.1 Register Map

The Keystone HyperLink contains 3 main sets of registers.

- Local HyperLink module configuration registers
- Remote HyperLink module configuration registers
- Local HyperLink SerDes configuration registers

Table 3-1 summarizes all 3 types of the HyperLink registers.

Table 3-1 Register Map (Part 1 of 2)

Address Offset	Register	Section
LOCAL HYPERLINK CONFIGURATION REGISTERS		Section 3.2
0x00	Revision/ID Register	Section 3.2.1
0x04	Control Register	Section 3.2.2
0x08	Status Register	Section 3.2.3
0x0C	Interrupt Priority Vector Status/Clear Register	Section 3.2.4
0x10	Interrupt Status/Clear Register	Section 3.2.5
0x14	Interrupt Pending/Set Register	Section 3.2.6
0x18	Generate Soft Interrupt Value	Section 3.2.7
0x1C	Tx Address Overlay Control	Section 3.2.8
0x20-28	Reserved	Reserved
0x2c	Rx Address Selector Control	Section 3.2.9
0x30	Rx Address PrivID Index	Section 3.2.10
0x34	Rx Address PrivID Value	Section 3.2.11
0x38	Rx Address Segment Index	Section 3.2.12
0x3c	Rx Address Segment Value	Section 3.2.13
0x40	Chip Version Register	Section 3.2.14
0x44	Lane Power Management Control	Section 3.2.15
0x48	Rx Priority Control (KeyStone II only)	Section 3.2.16
0x4C	ECC Error Counters	Section 3.2.17
0x50-0x54	Reserved	Reserved
0x58	Link Status Register	Section 3.2.18
0x5C	Side Band State (KeyStone II only)	Section 3.2.19
0x60	Interrupt Control Index	Section 3.2.20
0x64	Interrupt Control Value	Section 3.2.21
0x68	Interrupt Pointer Index	Section 3.2.22
0x6C	Interrupt Pointer Value	Section 3.2.23
0x70	SerDes Control and Status 1	Section 3.2.24
0x74	SerDes Control and Status 2	Section 3.2.25
0x78	SerDes Control and Status 3	Section 3.2.26
0x7C	SerDes Control and Status 4	Section 3.2.27
REMOTE HYPERLINK CONFIGURATION REGISTERS		Section 3.3
0x80	Remote Revision Register	Section 3.3
0x84	Remote Control Register	Section 3.3
0x88	Remote Status Register	Section 3.3
0x8C	Remote Interrupt Priority Vector Status/Clear Register	Section 3.3
0x90	Remote Interrupt Status/Clear Register	Section 3.3

Table 3-1 Register Map (Part 2 of 2)

Address Offset	Register	Section
0x94	Remote Interrupt Pending/Set Register	Section 3.3
0x98	Remote Generate Soft Interrupt Value	Section 3.3
0x9C	Remote Tx Address Overlay Control	Section 3.3
0xA0-0xA8	Reserved	Reserved
0xAC	Remote Rx Address Selector Control	Section 3.3
0xB0	Remote Rx Address PrivID Index	Section 3.3
0xB4	Remote Rx Address PrivID Value	Section 3.3
0xB8	Remote Rx Address Segment Index	Section 3.3
0xBC	Remote Rx Address Segment Value	Section 3.3
0xC0	Remote Chip Version Register	Section 3.3
0xC4	Remote Lane Power Management Control	Section 3.3
0xC8	Rx Priority Control (KeyStone II only)	Section 3.3
0xCC	Remote ECC Error Counters	Section 3.3
0xD0-0xD4	Reserved	Reserved
0xD8	Remote Link Status Register	Section 3.3
0xDC	Side Band State (KeyStone II only)	Section 3.3
0xE0	Remote Interrupt Control Index	Section 3.3
0xE4	Remote Interrupt Control Value	Section 3.3
0xE8	Remote Interrupt Pointer Index	Section 3.3
0xEC	Remote Interrupt Pointer Value	Section 3.3
0xF0	Remote SerDes Control and Status 1	Section 3.3
0xF4	Remote SerDes Control and Status 2	Section 3.3
0xF8	Remote SerDes Control and Status 3	Section 3.3
0xFC	Remote SerDes Control and Status 4	Section 3.3
HYPERLINK SERDES REGISTERS		Section 3.4
0x02620160	HyperLink SerDes Status Register	Section 3.4.1
0x026203B4	HyperLink SerDes PLL Configuration Register	Section 3.4.2
0x026203B8	HyperLink SerDes Receive Configuration Register 0	Section 3.4.3
0x026203BC	HyperLink SerDes Transmit Configuration Register 0	Section 3.4.4
0x026203C0	HyperLink SerDes Receive Configuration Register 1	Section 3.4.3
0x026203C4	HyperLink SerDes Transmit Configuration Register 1	Section 3.4.4
0x026203C8	HyperLink SerDes Receive Configuration Register 2	Section 3.4.3
0x026203CC	HyperLink SerDes Transmit Configuration Register 2	Section 3.4.4
0x026203D0	HyperLink SerDes Receive Configuration Register 3	Section 3.4.3
0x026203D4	HyperLink SerDes Transmit Configuration Register 3	Section 3.4.4
End of Table 3-1		

3.2 Local HyperLink Configuration Registers

They local HyperLink configuration registers are used to configure the HyperLink module on the local device. These registers are discussed in more detail below.

3.2.1 Revision Register (Base Address + 0x00)

The Revision Register contains the major and minor revisions for the HyperLink module.

Figure 3-1 Revision Register

31 30	29 28	27 16	15 11	10 8	7 6	5 0
Scheme	BU	Func	RTL Version	revmaj	Customer	revmin
R	R	R	R	R	R	R

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

Table 3-2 Revision Register Field Descriptions

Bits	Field	Type	Reset	Description
31-30	Scheme	R	01	Scheme – Used to distinguish current scheme
29-28	BU	R	00	Business Unit – Used to identify the creating business unit
27-16	Func	R	0xe90	Function identifier – 390 is for HyperLink
15-11	RTL Version	R	3	RTL Version identifies the version of the RTL. it is 14 for KeyStone II devices
10-8	revmaj	R	0x1	Major revision
7-6	Customer	R	00	Customer – Indicate a special version
5-0	revmin	R	0x0	Minor revision. it is 0x1 for KeyStone II devices

3.2.2 Control Register (Base Address + 0x04)

The Control Register determines operation of the HyperLink module.

Figure 3-2 Control Register

31 15	14	13	12 8	7	6 4	3	2	1	0
Reserved	intlocal	intenable	intvec	int2cfg	dataarbcyc	reserved	serial_stop	iloop	reset
R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

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

Table 3-3 Control Register Field Descriptions

Bits	Field	Type	Reset	Description
31-15	Reserved	R	0	Always read as 0. Writes have no affect.
14	intlocal	R/W	0	Interrupt Local. This bit determines whether interrupts are posted in the Interrupt Status/Clear Register or forwarded via the serial interface. When set, interrupts are posted in the Interrupt Status/Clear Register and the Hyperlink_intpls_o and Hyperlink_intlvl_o pins are asserted. When clear, interrupts are forwarded out the serial interface to the remote device.
13	intenable	R/W	0	Interrupt Enable. This bit causes HyperLink module status interrupts to be posted to the Interrupt Pending/Set Register (i.e. bit 7, 8 of Status Register).
12-8	intvec	R/W	0	Interrupt Vector. This field indicates which bit in the Interrupt Pending/Set Register is set for HyperLink module status interrupts.
7	int2cfg	R/W	0	Interrupt to Configuration Register. When this bit is set, the Interrupt Pending/Set Register is written directly with the status contained in interrupt packets. The least significant 8 bits of the Interrupt Pointer Register are used to point to the Interrupt Pending/Set Register. When clear, bits [31-2] of the Interrupt Pointer Register are used by the VBUSM Master interface as the address of interrupt Pending/Set register.

Table 3-3 Control Register Field Descriptions

Bits	Field	Type	Reset	Description
6-4	dataarbcyc	R/W	0	This field is used to allow local commands to momentarily take priority after dataarbcyc count of data bus words have been returned due to remote reads. it prevents deadlock from continuous sending local commands to remote device. if user sets zero, it is automatically converted to eight as a default value. (This field is only applied to KeyStone II devices)
3	Reserved	R	0	Always read as 0. Writes have no affect.
2	serial_stop	R/W	0	Serial_Stop. When it is set, it will disable all portal or remote register operation, and error them with a bad_address status. This bit should be set before iloop or reset bits are changed.
1	iloop	R/W	0	Hyperlink Internal loop-back. This bit when set causes the serial transmit data to be wrapped back to the serial receive data. When changing this bit, it is recommended that the *serial_stop* bit be set and that all outstanding transactions have completed before the *iloop* bit is changes at which time the *serial_stop* bit can be cleared to resume normal operation.
0	reset	R/W	0	Reset. When this bit is set, all internal state machines are reset, the serial interface is disabled, and link is lost. Note: Any bus transaction in flight between the devices will be lost. When changing this bit, it is recommended that the serial_stop bit be set and that all outstanding transactions have completed before the *reset* bit is set. After the *reset* bit is cleared the *serial_stop* bit can be cleared to resume normal operation.

3.2.3 Status Register (Base Address + 0x08)

The Status Register is used to detect conditions that may be of interest to the system designer.

Figure 3-3 Status Register

31 28	27 24	23 20	19 14
Reserved	swidthin	swidthout	Reserved
R	R	R	R

13	12	11	10	9	8	7	6	5	4	3	2	1	0
serial_halt	pll_unlock	rpend	iflow	oflow	rerror	lerror	nfempty3	nfempty2	nfempty1	nfempty0	spend	mpend	link
R	R	R	R	R	W/C	W/C	R	R	R	R	R	R	R

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

Table 3-4 Status Register Field Descriptions

Bits	Field	Type	Reset	Description
31-28	Reserved	R	0	Always read as 0. Writes have no affect.
27-24	swidthin	R	0	Size of the inbound serial data capability.
23-20	swidthout	R	0	Size of the outbound serial data capability.
19-14	Reserved	R	0	Always read as 0. Writes have no affect.
13	serial_halt	R	1	Indicates that the serial logic is in a halted state due to any of *reset*, *serial_stop*, *pll_unlock* being set.
12	pll_unlock	R	1	Indicates that the SerDes PLL is not locked to the reference clock. This will prevent any serial operations.
11	rpend	R	0	Remote Pending Request, indicates that a remote operation is currently pending or in flight. The user should monitor this bit after setting serial_stop and before changing iloop or reset register bits.
10	iflow	R	0	Inbound Flow Control. Indicates that a flow control enable request has been received and has stalled transmit until a flow control disable request is received.
9	oflow	R	0	Outbound Flow Control. Indicates that the internal flow control threshold has been reached and a flow control enable request has been sent to the remote device.
8	rerror	w/c	0	Remote error. This bit indicates that a downstream HyperLink module has detected an uncorrectable ECC error. This bit is set when an ECC status is received from the management interface. This bit is cleared by writing a one to it and the remote serial interface has been reset. When set, this bit will cause an interrupt if enabled in the Control Register Bit 13. The remote device will need to perform a serial reset or device reset to recover from this catastrophic failure. Since this indicates that a transaction has been lost, if the rpend bit is set along with this bit, it is possible that this device will also require a reset as a transaction from this device may have been lost.

Table 3-4 Status Register Field Descriptions

Bits	Field	Type	Reset	Description
7	lerror	w/c	0	Local error. This bit indicates that an inbound packet contains an uncorrectable ECC error. This bit is cleared by writing a one to it after the error has been corrected. When set, this bit will cause an interrupt if enabled in the Control Register Bit 13. This error indicate a catastrophic failure and that the receive link is down. The *reset* bit must be toggled to recover from this error. It is possible that returning transactions may have been lost in which case a device reset maybe necessary to recover.
6	nfempty3	R	0	FIFO 3 Not Empty. This bit indicates that the Slave Command is not empty.
5	nfempty2	R	0	FIFO 2 Not Empty. This bit indicates that the Slave Data FIFO is not empty.
4	nfempty1	R	0	FIFO 1 Not Empty. This bit indicates that the Master Command FIFO is not empty.
3	nfempty0	R	0	FIFO 0 Not Empty. This bit indicates that the Master Data FIFO is not empty.
2	spend	R	0	Pending slave requests. Indicates that a request has been detected on the Tx VBUSM slave interface.
1	mpend	R	0	Pending master requests. Indicates that a request has been asserted on the Rx VBUSM master interface.
0	link	R	0	Link. Indicates that the serial interface initialization sequence has completed successfully

3.2.4 Interrupt Priority Vector Status/Clear Register (Base Address + 0x0C)

When read, the Interrupt Priority Vector Status/Clear register displays the highest priority vector with a pending interrupt. When writing, only bits [4-0] are valid, and the value represents the vector of the interrupt to be cleared.

Figure 3-4 Interrupt Priority Vector Status/Clear Register

31	30 5	4 0
nointpend	Reserved	intstat
R	R	R/W

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

Table 3-5 Interrupt Priority Vector Status/Clear Register Field Descriptions

Bits	Field	Type	Reset	Description
31	nointpend	R	1	Interrupt pending. A value of 1 indicates there are no pending interrupts from the Interrupt Status/Clear Register. A value of 0 indicates there is a pending interrupt.
30-5	Reserved	R	0	Always read as 0. Writes have no affect.
4-0	intstat	R/W	0	When read, this field displays the vector that is mapped to the highest priority interrupt bit that is pending from the Interrupt Status/Clear Register, with Bit 0 as the highest priority, and Bit 31 as the lowest. Writing back the vector value into this field will clear the interrupt.

3.2.5 Interrupt Status/Clear Register (Base Address + 0x10)

The Interrupt Status/Clear Register indicates the unmasked interrupt status. Writing 1 to any bit in this register will clear the corresponding interrupt.

Figure 3-5 Interrupt Status/Clear Register

31 0
intclr
R/W

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

Table 3-6 Interrupt Status/Clear Register Field Descriptions

Bits	Field	Type	Reset	Description
31-0	intclr	R/W	0	This field indicates the unmasked status of each interrupt. Writing 1 to any bit in this field will clear the corresponding interrupt. The Hyperlink_int_o pin is driven high when any bit in this register is set.

3.2.6 Interrupt Pending/Set Register (Base Address + 0x14)

The Interrupt Pending/Set Register indicates the pending interrupt status. This register can be written by the local host or by remote interrupt packets when the int2cfg bit is set. When bits are set in this register, an interrupt output is signaled, if not already pending. Any write with a value of 0x0000 to this register will EOI the Interrupt Pending/Set Register interrupt output pins so they retrigger the interrupt. That is, writing a zero to the Interrupt Pending/Set Register retriggers the output interrupt lines if any bits are still set in this register.

Figure 3-6 Interrupt Pending/Set Register

31 0
intset
R/W

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

Table 3-7 Interrupt Pending/Set Register Field Descriptions

Bits	Field	Type	Reset	Description
31-0	intset	R/W	0	This field indicates the unmasked status of each pending interrupt. Writing any bit in this register will activate the interrupt output pin if not already active.

3.2.7 Generate Soft Interrupt Value Register (Base Address + 0x18)

The Generate Soft Interrupt Register should be written with a vector of the hardware index of the interrupt that is enabled for software interrupts. If the isec bit is also set, the csecure interface pin must be set to set the software interrupt. This register is also used to EOI Hyperlink_int_i hardware interrupts programmed in level mode.

Figure 3-7 Generate Soft Interrupt Value Register

31 16	15 8	7 0
Reserved	eoi_flag	ivector
R	W	R?W

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

Table 3-8 Generate Soft Interrupt Value Register Field Descriptions

Bits	Field	Type	Reset	Description
31-16	Reserved	R	0	Always read as 0. Writes have no affect.
15-8	eoi_flag	wo	0	Always read as zero, Writing 0xff to the eoi_flag along with 0xff to the ivector will EOI ALL level sensitive interrupt inputs, and writing 0xfe to the eoi_flag will EOI the particular ivector hardware interrupt input.
7-0	ivector	R/W	0	Ivector is the hardware index for the interrupt that will be set. If the SIEN bit of the Interrupt Control[IVECTOR] is set and the security level is met, the internal hardware pending bit will be set. This interrupt is forwarded based on the intlocal bit setting.

3.2.8 Tx Address Overlay Control Register (Base Address + 0x1c)

The Tx Address Map Mask Register is used to trim the transmitted packet address to remote device VBUSM addresses. See Section 2.7.4.1 "Address Translation on the TX Side" on page 2-20 for more information.

Figure 3-8 Tx Address Overlay Control Register

31 20	19 16	15 12	11 8	7 4	3 0
Reserved	txsecovl	Reserved	txprividovl	Reserved	txigmask
R	R/W	R	R/W	R	R/W

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

Table 3-9 TX Address Overlay Control Register Field Descriptions (Part 1 of 2)

Bits	Field	Type	Reset	Description
31-20	Reserved	R	0	Always read as 0. Writes have no affect.
19-16	txsecovl	R/W	0	Specifies where the Secure bit is placed in the outgoing TxAddress. This overlay occurs after the PrivID overlay by the txprividovl occurs. If txsecovl = 0, the secure bit is not overlayed. If txsecovl =1, secure bit is overlayed on TXAddress[17]. If txsecovl =2, secure bit is overlayed on TXAddress[18]. If txsecovl =3, secure bit is overlayed on TXAddress[19]. If txsecovl =4, secure bit is overlayed on TXAddress[20]. If txsecovl =5, secure bit is overlayed on TXAddress[21]. If txsecovl =6, secure bit is overlayed on TXAddress[22]. If txsecovl =7, secure bit is overlayed on TXAddress[23]. If txsecovl =8, secure bit is overlayed on TXAddress[24]. If txsecovl =9, secure bit is overlayed on TXAddress[25]. If txsecovl =10, secure bit is overlayed on TXAddress[26]. If txsecovl =11, secure bit is overlayed on TXAddress[27]. If txsecovl =12, secure bit is overlayed on TXAddress[28]. If txsecovl =13, secure bit is overlayed on TXAddress[29]. If txsecovl =14, secure bit is overlayed on TXAddress[30]. If txsecovl =15, secure bit is overlayed on TXAddress[31].
15-12	Reserved	R	0	Always read as 0. Writes have no affect.

Table 3-9 TX Address Overlay Control Register Field Descriptions (Part 2 of 2)

Bits	Field	Type	Reset	Description
11-8	txprividovl	R/W	0	Specifies where the PrivID is placed in the outgoing TxAddress. This overlay occurs after the incoming address is masked by the txigmask. If txprividovl = 0, no PrivID overlay If txprividovl = 1, TxAddr[20-17] = PrivID[3-0] If txprividovl = 2, TxAddr[21-18] = PrivID[3-0] If txprividovl = 3, TxAddr[22-19] = PrivID[3-0] If txprividovl = 4, TxAddr[23-20] = PrivID[3-0] If txprividovl = 5, TxAddr[24-21] = PrivID[3-0] If txprividovl = 6, TxAddr[25-22] = PrivID[3-0] If txprividovl = 7, TxAddr[26-23] = PrivID[3-0] If txprividovl = 8, TxAddr[27-24] = PrivID[3-0] If txprividovl = 9, TxAddr[28-25] = PrivID[3-0] If txprividovl = 10, TxAddr[29-26] = PrivID[3-0] If txprividovl = 11, TxAddr[30-27] = PrivID[3-0] If txprividovl = 12, TxAddr[31-28] = PrivID[3-0] If txprividovl = 13, TxAddr[31-29] = PrivID[2-0] If txprividovl = 14, TxAddr[31-30] = PrivID[1-0] If txprividovl = 15, TxAddr[31] = PrivID[0]
7-4	Reserved	R	0	Always read as 0. Writes have no affect.
3-0	txigmask	R/W	0	The txigmask is used to create the mask that is logically anded to the incoming address to create the address sent to the remote. If Txigmask=0, the mask = 0x0001FFFF; If Txigmask=1, the mask = 0x0003FFFF; If Txigmask=2, the mask = 0x0007FFFF; If Txigmask=3, the mask = 0x000FFFFF If Txigmask= 4, the mask = 0x001FFFFF; If Txigmask= 5, the mask = 0x003FFFFF If Txigmask= 6, the mask = 0x007FFFFF If Txigmask= 7, the mask = 0x00FFFFFF If Txigmask= 8, the mask = 0x01FFFFFF If Txigmask= 9, the mask = 0x03FFFFFF If Txigmask= 10, the mask = 0x07FFFFFF If Txigmask= 11, the mask = 0x0FFFFFFF If Txigmask= 12, the mask = 0x1FFFFFFF If Txigmask= 13, the mask = 0x3FFFFFFF If Txigmask= 14, the mask = 0x7FFFFFFF If Txigmask= 15, the mask = 0xFFFFFFFF

3.2.9 Rx Address Selector Control (Base Address + 0x2c)

The Rx Address Selector Control Register is used to configure which Rx Address bits select the Secure, PrivID and Segment/Length value arrays. This register also holds the secure value when the secure selection is one or zero. See Section 2.7.4.2 "Address Translation on the RX Side" on page 2-20 for more information.

Figure 3-9 Rx Address Selector Control Register

31 26	25	24	23 20	19 16	15 12	11 8	7 4	3 0
Reserved	rxsechi	rxseclo	Reserved	rxsecsel	Reserved	rxprividsel	Reserved	rxsegsel
R	R/W	R/W	R	R/W	R	R/W	R	R/W

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

Table 3-10 RX Address Selector Control Register Field Descriptions (Part 1 of 2)

Bits	Field	Type	Reset	Description
31-26	Reserved	R	0	Always read as 0. Writes have no affect.
25	rxsechi	R/W	0	The rxsechi bit is used as the value of the secure signal when the rxsecsel selected RxAddress is a one.
24	rxseclo	R/W	0	The rxseclo bit is used as the value of the secure signal when the rxsecsel selected RxAddress is a zero.
23-20	Reserved	R	0	Always read as 0. Writes have no affect.
19-16	rxsecsel	R/W	0x0	The rxsecsel value is used to select which bit of the RxAddress are used to select the rxsechi or rxseclo value used as the output secure signal value. If the Selected RxAddress is high, the output secure signal is from the rxsechi register field. If the Selected RxAddress is low, the output secure signal is from the rxseclo register field. To disable this function, set both rxsechi and rxseclo to the same value required. If rxsecsel=0, the selector bit = RxAddress [16] If rxsecsel=1, the selector bit = RxAddress [17] If rxsecsel=2, the selector bit = RxAddress [18] If rxsecsel=3, the selector bit = RxAddress [19] If rxsecsel=4, the selector bit = RxAddress [20] If rxsecsel=5, the selector bit = RxAddress [21] If rxsecsel=6, the selector bit = RxAddress [22] If rxsecsel=7, the selector bit = RxAddress [23] If rxsecsel=8, the selector bit = RxAddress [24] If rxsecsel=9, the selector bit = RxAddress [25] If rxsecsel=10, the selector bit = RxAddress [26] If rxsecsel=11, the selector bit = RxAddress [27] If rxsecsel=12, the selector bit = RxAddress [28] If rxsecsel=13, the selector bit = RxAddress [29] If rxsecsel=14, the selector bit = RxAddress [30] If rxsecsel=15, the selector bit = RxAddress [31]
15-12	Reserved	R	0	Always read as 0. Writes have no affect.

Table 3-10 RX Address Selector Control Register Field Descriptions (Part 2 of 2)

Bits	Field	Type	Reset	Description
11-8	rxprividsel	R/W	0x0	The rxprividsel value is used to select which bits of the RxAddress are used to select which PrivID array element value to drive to the PrivID signal. If rxprividsel = 0; use 0 as privID lookup index If rxprividsel = 1; use RxAddress [20-17]as privID lookup index If rxprividsel = 2; use RxAddress [21-18] as privID lookup index If rxprividsel = 3; use RxAddress [22-19] as privID lookup index If rxprividsel = 4; use RxAddress [23-20] as privID lookup index If rxprividsel = 5; use RxAddress [24-21] as privID lookup index If rxprividsel = 6; use RxAddress [25-22] as privID lookup index If rxprividsel = 7; use RxAddress [26-23] as privID lookup index If rxprividsel = 8; use RxAddress [27-24] as privID lookup index If rxprividsel = 9; use RxAddress [28-25] as privID lookup index If rxprividsel = 10; use RxAddress [29-26] as privID lookup index If rxprividsel = 11; use RxAddress [30-27] as privID lookup index If rxprividsel = 12; use RxAddress [31-28] as privID lookup index If rxprividsel = 13; use RxAddress [31-29] as privID lookup index If rxprividsel = 14; use RxAddress [31-30] as privID lookup index If rxprividsel = 15, use RxAddress [31] as privID lookup index
7-4	Reserved	R	0	Always read as 0. Writes have no affect.
3-0	rxsegsel	R/W	0x0	The rxsegsel value is used to select which bits of the RxAddress are used to select which Segment/Length array element value to Add/Check to the Address signal. If rxsegsel=0,use 0 as index to lookup segment/length table, use 0xffffffff as offset mask. If rxsegsel=1,use RxAddress [22-17] as index to lookup segment/length table, use 0x0001ffff as offset mask. If rxsegsel=2,use RxAddress [23-18]as index to lookup segment/length table, use 0x0003ffff as offset mask. If rxsegsel=3,use RxAddress [24-19]as index to lookup segment/length table, use 0x0007ffff as offset mask. If rxsegsel=4,use RxAddress [25-20]as index to lookup segment/length table, use 0x000fffff as offset mask. If rxsegsel=5,use RxAddress [26-21]as index to lookup segment/length table, use 0x001fffff as offset mask. If rxsegsel=6,use RxAddress [27-22]as index to lookup segment/length table, use 0x003fffff as offset mask. If rxsegsel=7, use RxAddress [28-23]as index to lookup segment/length table, use 0x007fffff as offset mask. If rxsegsel=8, use RxAddress [29-24]as index to lookup segment/length table, use 0x00ffffff as offset mask. If rxsegsel=9, use RxAddress [30-25]as index to lookup segment/length table, use 0x01ffffff as offset mask. If rxsegsel=10, use RxAddress [31-26]as index to lookup segment/length table, use 0x03ffffff as offset mask. If rxsegsel=11, use RxAddress [31-27]as index to lookup segment/length table, use 0x07ffffff as offset mask If rxsegsel=12, use RxAddress [31-28]as index to lookup segment/length table, use 0x0fffffff as offset mask If rxsegsel=13, use RxAddress [31-29]as index to lookup segment/length table, use 0x1fffffff as offset mask If rxsegsel=14, use RxAddress [31-30]as index to lookup segment/length table, use 0x3fffffff as offset mask If rxsegsel=15, use RxAddress [31]as index to lookup segment/length table, use 0x7fffffff as offset mask

3.2.10 Rx Address PrivID Index (Base Address + 0x30)

The Rx Address PrivID Index Register is used to select which PrivID array element is written or read from the Rx Address PrivID Value Register.

See Section 2.7.4.2.1 "PrivID Mapping on the RX Side" on page 2-20 for more information.

Figure 3-10 Rx Address PrivID Index Register

31 4	3 0
Reserved	rxprivid_idx
R	R/W

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

Table 3-11 Rx Address PrivID Index Register Field Descriptions

Bits	Field	Type	Reset	Description
31-4	Reserved	R	0	Always read as 0. Writes have no affect.
3-0	rxprivid_idx	R/W	0x0	The rxprivid_idx selects which PrivID array element is read or written using the Rx Address PrivID Value Register.

3.2.11 Rx Address PrivID Value Register (Base Address + 0x34)

The Rx Address PrivID Value Register is used to write or read the current value of a PrivID array element. See Section 2.7.4.2.1 "PrivID Mapping on the RX Side" on page 2-20 for more information.

Figure 3-11 Rx Address PrivID Value Register

31 4	3 0
Reserved	rxprivid_val
R	R/W

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

Table 3-12 Rx Address PrivID Value Register Field Descriptions

Bits	Field	Type	Reset	Description
31-4	Reserved	R	0	Always read as 0. Writes have no affect.
3-0	rxprivid_val	R/W	0x0	The rxprivid_val is the PrivID value read or written from the PrivID array element indexed by The Rx Address PrivID Index Register.

3.2.12 Rx Address Segment Index Register (Base Address + 0x38)

The Rx Address Segment Index Register is used to select which Segment/Length array element is read or written using the Rx Address Segment Value Register. See Section 2.7.4.2.2 "Address Remapping on the RX Side" on page 2-21 for more information.

Figure 3-12 Rx Address Segment Index Register

31 6	5 0
Reserved	rxseg_idx
R	R/W

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

Table 3-13 RX Address Segment Index Register Field Descriptions

Bits	Field	Type	Reset	Description
31-6	Reserved	R	0	Always read as 0. Writes have no affect.
5-0	rxseg_idx	R/W	0x00	The rxseg_idx selects which Segment/Length array element is read or written using the Rx Address Segment Value Register.

3.2.13 Rx Address Segment Value Register (Base Address + 0x3c)

The Rx Address Segment Value Register is used to read or write the selected Segment/Length array element. See Section 2.7.4.2.2 "Address Remapping on the RX Side" on page 2-21 for more information.

Figure 3-13 Rx Address Segment Value Register

31 16	15 5	4 0
rxseg_val	Reserved	rxlen_val
R/W	R	R/W

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

Table 3-14 Rx Address Segment Value Register Field Descriptions

Bits	Field	Type	Reset	Description
31-16	rxseg_val	R/W	0x0000	The rxseg_val is the Segment value read or written from the Segment/Length array element indexed by The Rx Address Segment Index Register.
15-5	Reserved	R	0	Always read as 0. Writes have no affect.
4-0	rxlen_val	R/W	0x00	The rxlen_val is the Length value read or written from the Segment/Length array element indexed by The Rx Address Segment Index Register. If rxlen_val = 0-7, the segment size is 0; If rxlen_val = 8, the segment size is 0x000000200 If rxlen_val = 9, the segment size is 0x000000400 If rxlen_val = 10, the segment size is 0x000000800 If rxlen_val = 11, the segment size is 0x000001000 If rxlen_val = 12, the segment size is 0x000002000 If rxlen_val = 13, the segment size is 0x000004000 If rxlen_val = 14, the segment size is 0x000008000If rxlen_val = 15, the segment size is 0x000010000 If rxlen_val = 16, the segment size is 0x000020000 If rxlen_val = 17, the segment size is 0x000040000 If rxlen_val = 18, the segment size is 0x000080000 If rxlen_val = 19, the segment size is 0x000100000 If rxlen_val = 20, the segment size is 0x000200000 If rxlen_val = 21, the segment size is 0x000400000 If rxlen_val = 22, the segment size is 0x000800000 If rxlen_val = 23, the segment size is 0x001000000 If rxlen_val = 24, the segment size is 0x002000000 If rxlen_val = 25, the segment size is 0x004000000 If rxlen_val = 26, the segment size is 0x008000000 If rxlen_val = 27, the segment size is 0x010000000 If rxlen_val = 28, the segment size is 0x020000000 If rxlen_val = 29, the segment size is 0x040000000 If rxlen_val = 30, the segment size is 0x080000000 If rxlen_val = 31, the segment size is 0x100000000

3.2.14 Chip Version Register (Base Address + 0x40)

The Chip Version Register reflects the value on the device_id and device_rev pins. This register provides a mechanism for software to determine the type and version of HyperLink devices. The value of device_id and device_rev field must be specified in the device specification. The device must be registered with the VLYNQ/HyperLink IP group to be assigned a unique ID to distinguish it from other VLYNQ or HyperLink devices.

Figure 3-14 Chip Version Register

31 16	15 0
devrev	devid
R	R

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

Table 3-15 Chip Version Register Field Descriptions

Bits	Field	Type	Reset	Description
31-16	devrev	R	device_rev	Device Revision. This field reflects the value of the device_rev pins.
15-0	devid	R	device_id	Device ID. This field reflects the value of the device_id pins. The device must be registered with the VLYNQ/HyperLink IP Group to be assigned an ID.

3.2.15 Lane Power Management Control Register (Base Address + 0x44)

The Power Management Control Register configures the modes of operation under which the HyperLink operates.

Figure 3-15 Lane Power Management Register

31 27	26 24	23 19	18 16	15 8	7 3	2	1	0
Reserved	H2L	Reserved	L2H	PWC	Reserved	quadlane	singlelane	zerolane
R	R/W	R	R/W	R/W	R	R/W	R/W	R/W

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

Table 3-16 Lane Power Management Register Field Descriptions

Bits	Field	Type	Reset	Description
31-27	Reserved	R	0x0	Always read as 0. Writes have no affect.
26-24	H2L	R/W	0x7	High to Low Clocks contains the Value that selects the number of clocks that the FIFO falls below ¼ rate before a transition from High to Low Speed is taken. If H2L=0, wait 64 clock cycles before transition If H2L=1, wait 128clock cycles before transition If H2L=2, wait 256clock cycles before transition If H2L=3, wait 512 clock cycles before transition If H2L=4, wait1024clock cycles before transition If H2L=5, wait 2048clock cycles before transition If H2L=6, wait 4096clock cycles before transition If H2L=7, wait 8192clock cycles before transition
23-19	Reserved	R	0x0	Always read as 0. Writes have no affect.

Table 3-16 Lane Power Management Register Field Descriptions

Bits	Field	Type	Reset	Description
18-16	L2H	R/W	0x7	Low to High Clocks contains the Value that selects the number of clocks that the FIFO is busy before a transition to High Speed is taken If L2H=0, wait 64 clock cycle before transition If L2H=1, wait 128clock cycle before transition If L2H=2, wait 256 clock cycle before transition If L2H=3, wait 512clock cycle before transition If L2H=4, wait 1024 clock cycle before transition If L2H=5, wait 2048 clock cycle before transition If L2H=6, wait 4096 clock cycle before transition If L2H=7, wait 8192 clock cycle before transition
15-8	PWC	R/W	0x00	Periodic Wakeup Control, enables periodic wake up of the SerDes, when set to 0, the SerDes will not be periodically woken up. This value plus one is the number of 65,536 SerDes clock tics between wake up events. At 12.5Gbps, a value of 1 is approximately 210uS. The expected operational value is currently 10mS (Value=94). When the timer expires, the SerDes will be woken up at full speed. If the SerDes is operated at full speed due to other reasons, the timer will start over. That is this controls the maximum interval between full speed events.
7-3	Reserved	R	0x0	Always read as 0. Writes have no affect.
2	quadlane	R/W	0x1	Quad Lane Enable. This bit enables or disables the ability to operate in four lane mode. Setting this bit to one enables four lane capability. Setting this bit to zero disable four lane mode. [1]
1	singlelane	R/W	0x1	Single Lane Enable. This bit enables or disables the ability to operate in single lane mode. Setting this bit to one enables single lane capability. Setting this bit to zero disable single lane mode.[1]
0	zerolane	R/W	0x1	Zero Lane. This bit enables or disables the ability to shut down to zero lanes. That is stop all SerDes operations and power down the SerDes. Setting this bit to one enables zero lane dynamic operation capability. Setting this bit to zero disable the ability to disable the SerDes dynamically, the SerDes will remain on until reset.

[1] If quadlane and singlelane both zero, four lane mode is assumed. if both mode is one, HW will dynamically control the power based on the amount of traffic

3.2.16 Rx Priority Control (Base Address + 0x48)

The Rx Priority Control register allows the master priority to be controlled. The tran0xxx are the highest priority bus request, and the tran1xxx are the lower priority request. The epriority is used only in the bus infrastructure to determine the priority operations. The end point accesses use the priority field for the priority operations. The only requirements are

1)The tran0pri should be a lower numeric value than tran1pri value.

2)The tranXepri should be equal to or a lower numeric value than tranXpri value

Figure 3-16 Rx Priority Control

31	30 28	27	26 24	23 15	14 12	11	10 8	7 0
Reserved	tran0pri	reserved	tran1pri	reserved	tran0epri	reserved	tran1epri	Reserved
R	R/W	R	R/W	R	R/W	R	R/W	R

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

Table 3-17 Rx Priority Control Field Descriptions

Bits	Field	Type	Reset	Description
31	Reserved	R	0x0	Always read as 0. Writes have no affect.
30-28	tran0pri	R/W	0x0	The tran0pri sets the priority field for the bus when the priority 0 is received
27	Reserved	R	0x0	Always read as 0. Writes have no affect.
26-24	tran1pri	R/W	0x4	The tran1pri sets the priority field for the bus when the priority 1 is received
23-15	Reserved	R	0x0	Always read as 0. Writes have no affect.
14-12	tran0epri	R/W	0x0	The tran0pri sets the epriority field for the bus when the priority 0 is received. This effective priority is used by the bus infrastructure to prioritize request.

Table 3-17 Rx Priority Control Field Descriptions

Bits	Field	Type	Reset	Description
11	Reserved	R	0x0	Always read as 0. Writes have no affect.
10-8	tran1epri	R/W	0x4	The tran1pri sets the epriority field for the bus when the priority 1is received. This effective priority is used by the bus infrastructure to prioritize request.
7-0	Reserved	R	0x0	Always read as 0. Writes have no affect.

3.2.17 ECC Error Counters Register (Base Address + 0x4c)

The ECC Error Counter register counts the number of correctable single bit errors detected by the receive PLS as well as the number of detectable double bit errors. This value can be used to determine the integrity of the SerDes Rx signal. Writing to this register clears the current counts to zero.

Figure 3-17 ECC Error Counters

31 16	15 8	7 0
sgl_err_cor	Reserved	dbl_err_det
R/W	R	R/W

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

Table 3-18 ECC Error Counters Field Descriptions

Bits	Field	Type	Reset	Description
31-16	sgl_err_cor	R/W	0x0	The Single Error Correctable counter is incremented when a correctable error is detected by the Rx PLS layer. Writing any value to this register will clear this counter.
15-8	Reserved	R	0x0	Always read as 0. Writes have no affect.
7-0	dbl_err_det	R/W	0x0	The Double Error Detectable counter is incremented when a detectable two bit error is detected by the Rx PLS layer. This indicates that the receive channel signal is marginal. Writing any value to this register will clear this counter.

3.2.18 Link Status Register (Base Address + 0x58)

The Link status register is used to debug failed link conditions. It contains valuable information about the start of the link-state machines. It is used only to determine what might be causing the link failure. Because the source of this register can change quickly, this register updates only when a change has been detected and it is capable of transferring it to the bus clock domain.

Figure 3-18 Link Status

31 30	29 28	27 26	25	24	23 20	19 16
txpls_req	txpls_ack	txpm_req	tx_rsync	txplsok	tx_phy_en	txflow_sts
R	R	R	R	R	R	R

15 14	13 12	11 10	9	8	7 4	3 0
rxpls_req	rxpls_ack	rxpm_req	rx_lsync	rx_one_id	rx_phy_en	rx_phy_pol
R	R	R	R	R	R	R

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

Table 3-19 Link Status Field Descriptions

Bits	Field	Type	Reset	Description
31-30	txpls_req	R	0	Indicates how many Tx lanes are requested. 1X=four lanes, 01=one lane, 00=zero lane
29-28	txpls_ack	R	0	Indicates ho many Tx lanes are active. 1X=four lanes, 01=one lane, 00=zero lane
27-26	txpm_req	R	0	Indicates the state of the Tx power management sideband control state. 11=Four lane mode, 10=four lane mode going to or from zero lane mode, 01=one lane mode, 00=zero lane mode.
25	tx_rsync	R	0	Indicates that the remote device has synced to the transmit training sequence. This sync is only true during the link upstate and it is not expected to be held since the txpls_ack indicates the lane link state
24	txplsok	R	0	Indicates that the Tx PLS layer has linked to the remote device.
23-20	tx_phy_en	R	0	Indicates which SerDes lanes are enabled
19-16	txflow_sts	R	0	Indicates which Flow bit are set from the remote receiver. This version only uses bits [1-0].
15-14	rxpls_req	R	0	Indicates how many Rx lanes are requested. 10=four lanes, 01=one lane, 00=zero lane
13-12	rxpls_ack	R	0	Indicates how many Rx lanes are active, 1x=four lanes, 01=one lane, 00=zero lane.
11-10	rxpm_req	R	0	Indicates the state of the Rx power management sideband control state. 11=Four lane mode, 10=four lane mode going to or from zero lane mode, 01=one lane mode, 00=zero lane mode.
9	rx_lsync	R	0	Indicates that the receive has synced to the training sequence. This sync is only true during the link upstate and it is not expected to be held since the rxpls_ack indicates the lane link state
8	rx_one_id	R	0	Indicates that the receiver has identified during the first link training that it has identified which Rx lane carries the remote Tx lane 0 data, so that if a request to enable one lane occurs it would only activate the one receiver lane necessary to operate. This bit must be set for the receiver to enter one lane mode, but since this bit will be set at the completion of training of all lanes initiated by either exiting serial reset or module reset and PLL lock. This bit should stay high forever unless a serial or module reset occurs
7-4	rx_phy_en	R	0	Indicates which SerDes lanes are enabled
3-0	rx_phy_pol	R	0	Indicates which SerDes lanes are reversed polarity

3.2.19 Side Band State (Base Address + 0x5C)

The Side Band State displays the state of the side band signals for both the Rx and Tx side band interfaces.

Figure 3-19 Side Band State

31 24	23 20	19 16	15 8	7 4	3 0
txbcmd	txcap	txsts	rxbcmd	rxcap	rxsts
R	R	R	R	R	R

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

Table 3-20 Side Band State Field Descriptions

Bits	Field	Type	Reset	Description
31-24	txbcmd	R	0x0	The txbcmd indicates the remote bus commands received on the Tx side band interface. For bit0, 1 means channel0 flow controlled, 0 means channel0 is operational. For bit1, 1means channel1 flow controlled, 0 means channel 1 is operational. For bit 4, 1 means receive sync locked, 0 means receive sync is not locked. all other bits are reserved.
23-20	txcap	R	0x0	The txcap indicates the remote capabilities received on the Tx side band interface. bit0 means high speed capable, bit1 means four lane capable. bit2,3 is reserved
19-16	txsts	R	0x0	The txsts indicates the remote status received on the Tx side band interface. bit0 means the Remote lost ECC sync, bit1 means Remote station management link down. The TxFL/TxPM carries the state of the RxFL/RxPM station management link, and the RxFL/RxPM carries the state of the TxFL/TxPM station management link. bit2,3 is reserved

Table 3-20 Side Band State Field Descriptions

Bits	Field	Type	Reset	Description
15-8	rxbcmd	R	0x0	The rxbcmd indicates the remote bus commands received on the Rx side band interface. For bit0, 1 means channel0 flow controlled, 0 means channel0 is operational. For bit1, 1means channel1 flow controlled, 0 means channel 1 is operational. For bit 4, 1 means receive sync locked, 0 means receive sync is not locked. all other bits are reserved.
7-4	rxcap	R	0x0	The rxcap indicates the remote capabilities received on the Rx side band interface. bit0 means high speed capable, bit1 means four lane capable. bit2,3 is reserved
3-0	rxsts	R	0x0	The rxsts indicates the remote status received on the Rx side band interface. bit0 means the Remote lost ECC sync, bit1 means Remote station management link down. The TxFL/TxPM carries the state of the RxFL/RxPM station management link, and the RxFL/RxPM carries the state of the TxFL/TxPM station management link. bit2,3 is reserved

3.2.20 Interrupt Control Index (Base Address + 0x60)

The Interrupt Control Index Register is used to control which hardware or software internal control register is read or written via the Interrupt Control Value Register.

Figure 3-20 Interrupt Control Index

31 8	7 0
Reserved	icidx
R	R/W

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

Table 3-21 Interrupt Control Index Field Descriptions

Bits	Field	Type	Reset	Description
31-8	Reserved	R	0x0	Always read as 0. Writes have no affect.
7-0	icidx	R/W	0x0	Interrupt Control Index indicates which interrupt control register controller is written when reading or writing to the *Interrupt Control Value* register. the range of icidx is 0 ~ 63 and it is matched with Hyperlink_int_i input number or SW triggered interrupt to HW_EVENT

3.2.21 Interrupt Control Value (Base Address + 0x64)

The Interrupt Control Register reads or writes the associated fields to the appropriate icidx interrupt channel. All channels not supported will return zero and be unsettable.

Figure 3-21 Interrupt Control Value

31	30	29	28	27	26 18	17 16
Inten	Inttype	Intpol	isec	sien	Reserved	dnid
R/W	R/W	R/W	R/W	R/W	R	R/W

15 12	11 8	7 5	4 0
Reserved	mps	Reserved	vector
R	R/W	R	R/W

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

Table 3-22 Interrupt Control Value Field Descriptions

Bits	Field	Type	Reset	Description
31	Inten	R/W	0x0	Interrupt Enable. When set, this bit indicates that interrupts detected on the Hyperlink_int_i[icidx] input should be forwarded to the below dnid:mps:vector interrupt vector.
30	Inttype	R/W	0x0	Interrupt Type. When set, this bit indicates that the Hyperlink_int_i[icidx] interrupt is pulsed. When clear, this bit indicates that Hyperlink_int_i[icidx] is level sensitive.
29	Intpol	R/W	0x0	Interrupt Polarity When set, this bit indicates that the Hyperlink_int_i[icidx] interrupt is active low. When clear, this bit indicates that Hyperlink_int_i[icidx] is active high
28	isec	R/W	0x0	Interrupt Security indicates the security level that the master must have to set a software interrupt for this SoftInt[icidx].
27	sien	R/W	0	Software Interrupt Enable indicate if this interrupt can be issued via software writing to the *Generate Soft Interrupt* Value register for this SoftInt[icidx].
26-18	Reserved	R	0x0	Always read as 0. Writes have no affect.
17-16	dnid	R/W	0x0	Destination Network IDentifier (Currently not used)
15-12	Reserved	R	0x0	Always read as 0. Writes have no affect.
11-8	mps	R/W	0x0	MicroProcessor Select. this field indicates which micro processor (typically, interrupt controller within SoC) can be used to set interrupt when the outgoing interrupt packet is transferred to the remote device. maximum 8 unit can be chosen.
7-5	Reserved	R	0x0	Always read as 0. Writes have no affect.
4-0	vector	R/W	0x0	When the local device has int2Local =1, this field indicates which bit of interrupt pending register to set. When the local device has int2local = 0, this field is transferred to the remote device, which is used to indicate which bit of the interrupt pending register to set in the remote device.

3.2.22 Interrupt Pointer Index (Base Address + 0x68)

The Interrupt Pointer Index Register is used to control which Interrupt Pointer Register is read or written via the Interrupt Pointer Value Register. The Interrupt Pointer Registers typically map to microprocessor interrupt controller set registers which get set to a one to interrupt that processor.

Figure 3-22 Interrupt Pointer Index

31 4	3 0
Reserved	ipidx
R	R/W

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

Table 3-23 Interrupt Pointer Index Field Descriptions

Bits	Field	Type	Reset	Description
31-4	Reserved	R	0x0	Always read as 0. Writes have no affect.
3-0	ipidx	R/W	0x0	Interrupt Pointer Index controls which Interrupt Pointer Register controller is read or written via the Interrupt Pointer Value Register. the range of the ipidx is 0 ~ 7 and it is matched with the micro processor select (mps) number in the receiving interrupt packet

3.2.23 Interrupt Pointer Value (Base Address + 0x6c)

The Interrupt Pointer Value register is used to set the Interrupt Pointer (base of the interrupt controller set register) to which interrupts are written for the appropriate ipidx pointer register when the int2cfg bit is set to zero. All unsupported interrupt pointers will read zero and will be unsettable.

Figure 3-23 Interrupt Pointer Value

31 2	1 0
intptr	Reserved
R/W	R

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

Table 3-24 Interrupt Pointer Value Field Descriptions

Bits	Field	Type	Reset	Description
31-2	intptr	R/W	0x0	The Interrupt Pointer Value is write or read to the Selected (ipidx) Interrupt Pointer. This allows the remote interrupts to interrupt any MicroProcessor (typically, interrupt controller). Values written are stored in the Interrupt Pointer for that MicroProcessor. if int2cfg field in the Control register is set to one, only 8 LSB bits can be used to select register within the Hyperlink
1-0	Reserved	R	0x0	Always read as 0. Writes have no affect.

3.2.24 SerDes Control and Status 1 Register (Base Address + 0x70)

The SerDes Control and Status 1 Register is used to define the mask time that the receive lane data is ignored after enabling the lane(s) from either a sleep or disabled state. The default numbers of these counters are not yet determined. When these counters are zero, there are no delays in link establishment. This register delays the start of link establishment or step up link by a number of symbol times sixteen.

Figure 3-24 SerDes Control and Status 1 Register

31 24	23 16	15 0
sleep_cnt	disable_cnt	Reserved
R/W	R/W	R

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

Table 3-25 SerDes Control and Status 1 Register Field Descriptions

Bits	Field	Type	Reset	Description
31-24	sleep_cnt	R/W	0x09	SerDes sleep mask count, this count times 16 SerDes symbol times are masked for SerDes lanes that enter a sleep/enable state. This allows the internal SerDes power supplies to stabilize before the link is established.
23-16	disable_cnt	R/W	0x2E	SerDes disable mask count, this count times 16 SerDes symbol times are masked for SerDes lanes that enter a disabled state. This allows the SerDes CDR and equalizer to stabilize before the link is established.
15-0	Reserved	R	0x0	Always read as 0. Writes have no affect.

3.2.25 SerDes Control and Status 2 Register (Base Address + 0x74)

This register is Reserved for SerDes control and status operations. There is no defined functionality for this register.

Figure 3-25 SerDes Control and Status 2 Register

31 16	15 0
s2ctl	Reserved
R/W	R

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

Table 3-26 SerDes Control and Status 2 Register Field Descriptions

Bits	Field	Type	Reset	Description
31-16	s2ctl	R/W	0x0000	SerDes Control, has no defined function. Just a simple read write register.
15-0	Reserved	R	0x0	Always read as 0. Writes have no affect.

3.2.26 SerDes Control and Status 3 Register (Base Address + 0x78)

This register is Reserved for SerDes control and status operations. There is no defined functionality for this register.

Figure 3-26 Figure 2-25 SerDes Control and Status 3 Register

31 16	15 0
s3ctl	Reserved
R/W	R

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

Table 3-27 SerDes Control and Status 3 Register Field Descriptions

Bits	Field	Type	Reset	Description
31-16	s3ctl	R/W	0x0000	SerDes Control, has no defined function. Just a simple read write register.
15-0	Reserved	R	0x0	Always read as 0. Writes have no affect.

3.2.27 SerDes Control and Status 4 Register (Base Address + 0x7C)

The SerDes Control and Status Register is used to quicken DV so that the periodic wake-up timer can be tested more easily and to allow change to the SerDes non-runtime power levels. The power-level controls were added to reduce SerDes sleep mode functionality risk. It provides the ability to enable or disable sleep functionality for both the transmit and receive SerDes lanes.

Figure 3-27 Figure 2-26 SerDes Control and Status 4 Register

31	30 20	19 18	17 16	15 0
dvquick	s4ctl	tx_spc	rx_spc	Reserved
R/W	R/W	R/W	R/W	R

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

Table 3-28 SerDes Control and Status 4 Register Field Descriptions

Bits	Field	Type	Reset	Description
31	dvquick	R/W	0x0	DVQUICK reduces the periodic wake up event prescaler to 256 clocks instead of 65,536 clocks. It allow for more exhaustive testing of the PWC.
30-20	s4ctl	R/W	0x0000	SerDes Control, has no defined function. Just a simple read write register.
19-18	tx_spc	R/W	0	The transmit SerDes power control is used to modify the SLEEP and enable characteristics of the SerDes. 01=Full Sleep configuration, 00=fast sleep configuration, 1X=No sleep configuration. [1]
17-16	rx_spc	R/W	0	The receive SerDes power control is used to modify the SLEEP and enable characteristics of the SerDes. 01=Full Sleep configuration, 00=fast sleep configuration, 1X=No sleep configuration. [1]
15-0	Reserved	R	0x0	Always read as 0. Writes have no affect.

[1] These modes are added to reduce risk of new SerDes sleep features. In the event that the SerDes sleep functions do not operate as planned, they can be disabled for each the Rx and Tx SerDes interfaces. The 00 is the intended sleep configuration.

3.3 Remote HyperLink Configuration Registers

The Remote Configuration Registers are the same registers described above but for the remote HyperLink device. Configuration accesses are used to access these registers and do not require configuration of the address translation registers. These registers can be accessed at the location of the HyperLink base address + 0x80-0xFC.

3.4 HyperLink SerDes Configuration and Status Registers

This section describes the Keystone I HyperLink SerDes configuration and status registers.

Note—SerDes module information for KeyStone II devices is not provided in this user guide. Please check for availability of the SerDes User Guide for KeyStone II Devices on the device product page.

The Keystone I HyperLink SerDes configuration and status registers are shown in Table 3-29.

Table 3-29 KeyStone I HyperLink SerDes Configuration and Status Registers

Address	Register Mnemonic	Register Name	Section
0x02620160	HYPERLINK_SERDES_STS	HyperLink SerDes Status Register	Section 3.4.1
0x026203B4	HYPERLINK_SERDES_CFGPLL	HyperLink SerDes PLL Configuration Register	Section 3.4.2
0x026203B8	HYPERLINK_SERDES_CFGRX0	HyperLink SerDes Receive Configuration Register 0	Section 3.4.3
0x026203BC	HYPERLINK_SERDES_CFGTX0	HyperLink SerDes Transmit Configuration Register 0	Section 3.4.4
0x026203C0	HYPERLINK_SERDES_CFGRX1	HyperLink SerDes Receive Configuration Register 1	Section 3.4.3
0x026203C4	HYPERLINK_SERDES_CFGTX1	HyperLink SerDes Transmit Configuration Register 1	Section 3.4.4
0x026203C8	HYPERLINK_SERDES_CFGRX2	HyperLink SerDes Receive Configuration Register 2	Section 3.4.3
0x026203CC	HYPERLINK_SERDES_CFGTX2	HyperLink SerDes Transmit Configuration Register 2	Section 3.4.4
0x026203D0	HYPERLINK_SERDES_CFGRX3	HyperLink SerDes Receive Configuration Register 3	Section 3.4.3
0x026203D4	HYPERLINK_SERDES_CFGTX3	HyperLink SerDes Transmit Configuration Register 3	Section 3.4.4
End of Table 3-29			

3.4.1 HyperLink SerDes Status Register (HYPERLINK_SERDES_STS)

This section describes the HyperLink SerDes status register. This register contains the status information the HyperLink SerDes PLL, as well as all of the HyperLink SerDes receive and transmit lanes.

Figure 3-28 HyperLink SerDes Status Register

31 28	27	26	25	24
Reserved	EQOVER3	EQUNDER3	OCIP3	LOSDTCT3
R-0	R-0	R-0	R-0	R-0

23	22 21	20	19	18	17	16
SYNC3	Reserved	EQOVER2	EQUNDER2	OCIP2	LOSDTCT2	SYNC2
R-0	R-0	R-0	R-0	R-0	R-0	R-0

15 14	13	12	11	10	9	8
Reserved	EQOVER1	EQUNDER1	OCIP1	LOSDTCT1	SYNC1	Reserved
R-0	R-0	R-0	R-0	R-0	R-0	R-0

7	6	5	4	3	2	1	0
Reserved	EQOVER0	EQUNDER0	OCIP0	LOSDTCT0	SYNC0	Reserved	L:ock
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; -x, value is indeterminate — see the device-specific data manual

Table 3-30 HyperLink SerDes Status Register Field Descriptions

Bit	Field	Description
31-28	Reserved	Reserved.
27	EQOVER3	*Received signal over equalized.* Driven high asynchronously during equalizer analysis if the received signal is over equalized on SerDes receive lane 3.
26	EQUNDER3	*Received signal under equalized.* Driven high asynchronously during equalizer analysis if the received signal is under equalized on SerDes receive lane 3.
25	OCIP3	*Offset compensation in progress.* Driven high asynchronously during offset compensation on SerDes receive lane 3.
24	LOSDTCT3	*Loss of Signal detect.* Driven high asynchronously when a loss of signal condition is detected for receive lane 3.
23	SYNC3	*Symbol alignment.* When comma detection is enabled, this output is high when an aligned comma is received, in the same cycle that the comma pattern is output on SerDes lane 3. Alternatively, when an alignment jog is requested, it is high to indicate that the request has been completed.
22-21	Reserved	Reserved
20	EQOVER2	*Received signal over equalized.* Driven high asynchronously during equalizer analysis if the received signal is over equalized on SerDes receive lane 2.
19	EQUNDER2	*Received signal under equalized.* Driven high asynchronously during equalizer analysis if the received signal is under equalized on SerDes receive lane 2.
18	OCIP2	*Offset compensation in progress.* Driven high asynchronously during offset compensation on SerDes receive lane 2.
17	LOSDTCT2	*Loss of Signal detect.* Driven high asynchronously when a loss of signal condition is detected for receive lane 2.
16	SYNC2	*Symbol alignment.* When comma detection is enabled, this output is high when an aligned comma is received, in the same cycle that the comma pattern is output on SerDes lane 2. Alternatively, when an alignment jog is requested, it is high to indicate that the request has been completed.
15-14	Reserved	Reserved.
13	EQOVER1	*Received signal over equalized.* Driven high asynchronously during equalizer analysis if the received signal is over equalized on SerDes receive lane 1.
12	EQUNDER1	*Received signal under equalized.* Driven high asynchronously during equalizer analysis if the received signal is under equalized on SerDes receive lane 1.
11	OCIP1	*Offset compensation in progress.* Driven high asynchronously during offset compensation on SerDes receive lane 1.
10	LOSDTCT1	*Loss of Signal detect.* Driven high asynchronously when a loss of signal condition is detected for receive lane 1.
9	SYNC1	*Symbol alignment.* When comma detection is enabled, this output is high when an aligned comma is received, in the same cycle that the comma pattern is output on SerDes lane 1. Alternatively, when an alignment jog is requested, it is high to indicate that the request has been completed.
8-7	Reserved	Reserved.
6	EQOVER0	*Received signal over equalized.* Driven high asynchronously during equalizer analysis if the received signal is over equalized on SerDes receive lane 0.
5	EQUNDER0	*Received signal under equalized.* Driven high asynchronously during equalizer analysis if the received signal is under equalized on SerDes receive lane 0.
4	OCIP0	*Offset compensation in progress.* Driven high asynchronously during offset compensation on SerDes receive lane 0.
3	LOSDTCT0	*Loss of Signal detect.* Driven high asynchronously when a loss of signal condition is detected for receive lane 0.
2	SYNC0	*Symbol alignment.* When comma detection is enabled, this output is high when an aligned comma is received, in the same cycle that the comma pattern is output on SerDes lane 0. Alternatively, when an alignment jog is requested, it is high to indicate that the request has been completed.
1	Reserved	Reserved.
0	LOCK	*PLL Lock.* Driven high asynchronously between 2048-3071 refclk cycles after the PLL has locked, which will occur within 200 refclk cycles.
End of Table 3-30		

3.4.2 HyperLink SerDes PLL Configuration Register (HYPERLINK_SERDES_CFGPLL)

This section describes the HyperLink SerDes PLL configuration register. The main purpose of this register is to configure operating rate of the HyperLink SerDes based on the SerDes reference clock (refclk) and the multiplication factor (MPY). This register also provides other settings to improve the integrity of the signal. All of the HyperLink SerDes lanes share this configuration register. The HyperLink SerDes PLL configuration register is shown in Figure 3-29 and described in Table 3-28.

Note—The RATE bits, located in the SerDes RX configuration and TX configuration registers, also affect the operating rate of the HyperLink SerDes. See Section 3.4.3 and Section 3.4.4 for more information.

Figure 3-29 HyperLink SerDes PLL Configuration Register

31 13	12 11	10	9	8 1	0
Reserved	LOOP_BANDWIDTH	Reserved	VRANGE	MPY	ENPLL
R-0	R/W	R-0	R/W	R/W	R/W

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

Table 3-31 HyperLink SerDes PLL Configuration Register Field Descriptions (Part 1 of 2)

Bit	Field	Description
31-13	Reserved	Reserved.
12-11	LOOP_BANDWIDTH	Loop bandwidth. Specify loop bandwidth settings. Jitter on the reference clock input to the PLL is translated into the PLL output and impairs the ability of the transmitter and receiver to work properly. The loop bandwidth setting allows The user to select the setting that will reduce the jitter as much as possible in the PLL output. For most systems, medium bandwidth will be the best setting. All systems should start with medium bandwidth, and move to other values only when required to improve signal integrity. **00b = Medium Bandwidth (Recommended)** 01b = Ultra High Bandwidth 10b= Low Bandwidth 11b= High Bandwidth
10	Reserved.	Reserved.
9	VRANGE	Voltage Controlled Oscillator Range. This bit should be set according to the equation below. 0 = LINERATE × RATESCALE > 2.17GHz 1 = LINERATE × RATESCALE < 2.17GHz

Table 3-31 HyperLink SerDes PLL Configuration Register Field Descriptions (Part 2 of 2)

Bit	Field	Description
8-1	MPY	PLL multiply. Selects PLL multiply factors between 4 and 25. 00010000b = 4x 00010100b = 5x 00110000b = 6x 00100000b = 8x 00101000b = 10x 00110000b = 12x 00110010b = 12.5x 00111100b = 15x 00111100b = 16x 01000001b = 16.5x 01010000b = 20x 01011000b = 22x 01100100b = 25x
0	ENPLL	Enable PLL. This value is set automatically by the HyperLink module based on the RESET bit in the CONTROL register. 0 = PLL Disabled 1 = PLL Enabled
End of Table 3-31		

3.4.3 HyperLink SerDes Receive Configuration Register (HYPERLINK_SERDES_CFGRXn)

This section describes the HyperLink SerDes receive configuration register. The main purpose of this register is to configure HyperLink SerDes receiver. Each HyperLink receive lane has its own SerDes receive configuration register. The HyperLink SerDes receive configuration register is shown in Figure 3-30 and described in Table 3-32.

Figure 3-30 HyperLink SerDes RX Configuration Register n

31 — 25	24 — 23	22	21	20 — 18	17 — 15
Reserved	LOOPBACK	ENOC	EQHLD	EQ	CDR
R-0	R/W	R/W	R/W	R/W	R/W

14 — 12	11 — 10	9 — 7	6	5 — 4	3 — 1	0
LOS	ALIGN	TERM	INVPAIR	RATE	BUSWIDTH	ENRX
R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Table 3-32 HyperLink SerDes RX Configuration Register n Field Descriptions (Part 1 of 2)

Bit	Field	Description
31-25	Reserved	Reserved.
24-23	LOOPBACK	*Loopback*. Enables or disables loopback. This field is automatically set by the HyperLink module based on the iLoop bit in the CONTROL register.
22	ENOC	*Enable offset compensation*. Always set to 1 to enable offset compensation.
21	EQHLD	*Hold equalizer*. Holds the equalizer in its current state. Always set to 0 to enable equalizer adaptation and analysis.
20-18	EQ	*Equalizer*. Enables and configures the adaptive equalizer to compensate for loss in the transmission media. Always set to 001b ton enable fully adaptive equalization. This allows the low frequency gain of the equalizer to be determined algorithmically by analyzing the data patterns and transition positions in the received data. 001b = Fully adaptive equalization All other values are reserved.

Table 3-32 HyperLink SerDes RX Configuration Register n Field Descriptions (Part 2 of 2)

Bit	Field	Description
17-15	CDR	*Clock/data recovery*. This field configures the clock recovery algorithm that is used to adjust the clocks that sample the receive data. This allows slight variations in the clock frequency to be tracked and adjusted to ensure that the data is sampled midway between data transitions. First order and second order algorithms are provided, each with different tracking rates. Higher order algorithms and higher precision tracking rates will result in higher power consumption, so the tradeoff of phase tracking versus power consumption will need to be evaluated based on the requirements of each system. 000b = Second order tracking algorithm with +/-313ppm tracking rate. 001b = Second order tracking algorithm with +/-607ppm tracking rate. 010b = Second order tracking algorithm with +/-723ppm tracking rate. 011b = Second order tracking algorithm with +/-868ppm tracking rate. 100b = First order tracking algorithm with +/- 96ppm tracking rate. **101b = First order tracking algorithm with +/- 289ppm tracking rate. (Recommended)** 110b = First order tracking algorithm with +/- 434ppm tracking rate. 111b = First order tracking algorithm with +/- 13ppm tracking rate.
14-12	LOS	*Loss of signal detection*. Enables loss of signal detection with 2 selectable thresholds. 000b = Loss of signal detection disabled. The HyperLink module will automatically disable loss of signal detection when the iLoop bit is set in the CONTROL register. 100b = Loss of signal detection is enabled. All other values are reserved
11-10	ALIGN	*Symbol alignment*. This field is set automatically by the HyperLink module.
9-7	TERM	*Input termination*. Selects input termination options for AC or DC coupled systems. 001b = AC coupled systems. 111b= DC coupled systems. All other values are reserved.
6	INVPAIR	*Invert polarity*. Inverts polarity of RXp and RXn lines. This field is set automatically by the HyperLink module. 0 = Normal polarity. 1 = Inverted polarity.
5-4	RATE	Operating rate. Selects full, half, or quarter rate operation. 00b = Full rate. Four data samples are taken per PLL output clock cycle. 01b = Half rate. Two data samples are taken per PLL output clock cycle. 10b = Quarter rate. One data sample is taken every PLL output clock cycles. 11b = Eighth rate. One data sample is taken every 2 PLL output clock cycles.
3-1	BUSWIDTH	*Bus width*. This field is automatically set to 010b by the HyperLink module.
0	ENRX	*Enable receiver*. This field is set automatically by the HyperLink module. 0 = The receiver is disabled. 1 = The receiver is enabled.
End of Table 3-32		

3.4.4 HyperLink SerDes Transmit Configuration Register (HYPERLINK_SERDES_CFGTXn)

This section describes the HyperLink SerDes transmit configuration register. This register allows the user configure the HyperLink SerDes transmitter. Each HyperLink transmit lane has its own SerDes transmit configuration register. The HyperLink SerDes transmit configuration register is shown in Figure 3-31 and described in Table 3-33.

Figure 3-31 HyperLink SerDes TX Configuration Register n (Part 1 of 2)

31 23	22 21	20	19	18 14
Reserved	LOOPBACK	MSYNC	FIRUPT	TWPST1
R/W	R/W	R/W	R/W	R/W

Figure 3-31 HyperLink SerDes TX Configuration Register n (Part 2 of 2)

13 11	10 7	6	5 4	3 1	0
TWPRE	SWING	INVPAIR	RATE	BUSWIDTH	ENTX
R/W	R/W	R/W	R/W	R/W	R/W

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

Table 3-33 HyperLink SerDes TX Configuration Register n Field Descriptions (Part 1 of 2)

Bit	Field	Description
31-23	Reserved	Reserved.
22-21	LOOPBACK	*Loopback*. Enables loopback. For normal mode (non-loopback), this field is set automatically by the HyperLink module based on the value in the iLoop bit in the CONTROL register For loopback mode, bit 21 is automatically set by the HyperLink module based on the value in the iLoop bit in the CONTROL register. Bit 22 must be set by the user, and controls whether or not the power should be enabled to the TX pins. 00b = Loopback is disabled. This configuration is set automatically by the HyperLink module. 10b = Loopback enabled. 11b = Loopback enabled. All other values are reserved.
20	MSYNC	*Synchronization master*. Enables the channel as the master lane for synchronization purposes. This field is set to 1 automatically by the HyperLink module.
19	FIRUPT	*Transmitter pre and post cursor FIR filter update*. Update control of FIR tap weights. fields TWPRE and TWPST1 can be updated when SerDes byte clock and this input are both high. This field should always be set to 1.
18-14	TWPST1	*Adjacent post cursor Tap weight*. Selects one of 32 output tap weights for transmit waveform conditioning. The settings range from 37.5 to -37.5% in 2.5% steps. The best value for this field depends on a variety of factors, including trace length. For best results, it is recommended that users begin with one of the following 2 settings, and then adjusting the value based on the characteristics of the system. • For traces with 4” to 10” between devices, program this field to -27.5% (11011b). • For traces with less than 4” between devices, program this field to -17.5% (10111b) See Table 3-34 for the full list of tap weights available..
13-11	TWPRE	*Precursor Tap weight*. Selects one of 8 output tap weights for TX waveform conditioning. The settings range from 0 to -17.5% in 2.5% steps. The best value for this field depends on a variety of factors, including trace length. For best results, it is recommended that users begin by setting this field to -10% (100b), and then adjusting the value based on the characteristics of the system. 000b = 0% 001b = -2.5% 010b = -5.0% 011b = -7.5% 100b = -10.0% **(Recommended starting value)** 101b = -12.5% 110b = -15.0% 111b = -17.5%

Table 3-33 HyperLink SerDes TX Configuration Register n Field Descriptions (Part 2 of 2)

Bit	Field	Description
10-7	SWING	*Output swing*. Selects one of 16 output amplitude settings between 760 and 1220 mV$_{dfpp}$. The “best” value for this field will depend on several environmental factors, such as the board design, trace length, and temperature. The higher the value that is programmed into this field, the higher the signal amplitude (eye height). Higher amplitudes will make it easier for the receiver to interpret the signal; however higher amplitudes also correspond to higher power consumption. It is recommended that users start by programming 0111b to this register. If the receiver is not able to interpret the signal, then the value in this register can be increased to improve the signal integrity. If the receiver is able to understand the signal then a lower value can be programmed into this register to reduce power consumption. 0000b = 760 mV$_{dfpp}$ 0001b = 795 mV$_{dfpp}$ 0010b = 835 mV$_{dfpp}$ 0011b = 870 mV$_{dfpp}$ 0100b = 905 mV$_{dfpp}$ 0101b = 935 mV$_{dfpp}$ 0110b = 970 mV$_{dfpp}$ 0111b = 1005 mV$_{dfpp}$ **(Recommended starting configuration)** 1000b = 1035 mV$_{dfpp}$ 1001b = 1070 mV$_{dfpp}$ 1010b = 1100 mV$_{dfpp}$ 1011b = 1130 mV$_{dfpp}$ 1100b = 1155 mV$_{dfpp}$ 1101b = 1180 mV$_{dfpp}$ 1110b = 1200 mV$_{dfpp}$ 1111b = 1220 mV$_{dfpp}$
6	INVPAIR	*Invert polarity*. Inverts polarity of RXp and RXn lines. This field is set automatically by the HyperLink module. 0 = Normal polarity. 1 = Inverted polarity.
5-4	RATE	Operating rate. Selects full, half, or quarter rate operation. 00b = Full rate. Four data samples are taken per PLL output clock cycle. 01b = Half rate. Two data samples are taken per PLL output clock cycle. 10b = Quarter rate. One data sample is taken every PLL output clock cycles. 11b = Eighth rate. One data sample is taken every 2 PLL output clock cycles.
3-1	BUSWIDTH	*Bus width*. Always write 010b to this field to indicate a 20-bit wide parallel bus to the clock. All other values are reserved.
0	ENTX	*Enable Transmitter*. This field is set automatically by the HyperLink module. 0 = The transmitter is disabled. 1 = The transmitter is enabled.
End of Table 3-33		

Table 3-34 SerDes Transmit Configuration Post-Cursor Tap Weights (TWPST1) (Part 1 of 2)

Value	Positive Tap Weight (%)	Value	Negative Tap Weight (%)
00000b	0	10000b	0
00001b	+2.5	10001b	-2.5
00010b	+5.0	10010b	-5.0
00011b	+7.5	10011b	-7.5
00100b	+10.0	10100b	-10.0
00101b	+12.5	10101b	-12.5
00110b	+15.0	10110b	-15.0
00111b	+17.5	10111b	-17.5
01000b	+20.0	11000b	-20.0
01001b	+22.5	11001b	-22.5
01010b	+25.0	11010b	-25.0
01011b	+27.5	11011b	-27.5

Table 3-34 SerDes Transmit Configuration Post-Cursor Tap Weights (TWPST1) (Part 2 of 2)

Value	Positive Tap Weight (%)	Value	Negative Tap Weight (%)
01100b	+30.0	11100b	-30.0
01101b	+32.5	11101b	-32.5
01110b	+35.0	11110b	-35.0
01111b	+37.5	11111b	-37.5
End of Table 3-34			

Appendix A

HyperLink SerDes Programming Examples

A.1 KeyStone I Device SerDes Programming

The following shows an example configuration for the KeyStone I HyperLink SerDes registers. This example configures the Local HyperLink SerDes registers for communication with a Remote HyperLink device. All 4 lanes of the HyperLink SerDes are configured for 12.5 GHz operation. This configuration assumes that the SerDes REFCLK is operating at 312.5 MHz.

```
/* Unlock device control configuration area */
KICK0 = 0x83e70b13;
KICK1 = 0x95a4f1e0;

/***************************************************************************
* Configure HyperLink SerDes Receive Registers

* CFGRX [24:23] LOOPBACK = 0 (Automatically configured by HyperLink hardware)
* CFGRX [22] ENOC = 1
* CFGRX [21] EQHLD = 0
* CFGRX [20:18] EQ = 001b (fully adaptive)
* CFGRX [17:15] CDR = 101b (first order)
* CFGRX [14:12] LOS = 100b (Normal Mode: user must set to 100b)
* (Loopback Mode: LOS automatically set to 000b by HyperLink hardware)
* CFGRX [11:10] ALIGN = 00b (No alignment. Tied to zero)
* CFGRX [9:7] TERM =  001b (001b for AC coupling and 101b for DC coupling)
* CFGRX [6] INVPAIR = 0
* CFGRX [5:4] RATE = 00b full rate
* (4 bit per clock) 3.125GHz * 4 = 12.5Gbps max speed per lane
* CFGRX [3:1] BUSWIDTH = 10b (Fixed to 20 bit mode)
* CFGRX [0] ENRX = 1 (Automatically configured by the HyperLink hardware)
**************************************************************************/
HYPERLINK_SERDES_CFGRX0 = 0x0046C085;
HYPERLINK_SERDES_CFGRX1 = 0x0046C085;
HYPERLINK_SERDES_CFGRX2 = 0x0046C085;
HYPERLINK_SERDES_CFGRX3 = 0x0046C085;

/***************************************************************************
* Configure HyperLink SerDes Transmit Registers
* CFGTX [22:21] LOOPBACK = 00b (Automatically configured by the HyperLink hardware)
* CFGTX [20] MSYNC = 1 (Automatically configured by the HyperLink hardware)
* CFGTX [19] FIRUPT = 1   Transmitter pre and post cursor FIR filter update
* CFGTX [18:14] TWPST1 = 11011b Adjacent Post Cursor Tap Weight
* If trace length is 4” or less, use 23 (-17.5%).
* If trace length is between 4” and 10”, use 27 (-27.5%).
* CFGTX [13:11] TWPRE = 100b Precursor Tap weight (-10%)
* CFGTX [10:7] SWING = 1111b  7 for short trace length and 15 for long trace length
* CFGTX [6] INVPAIR = 0
* CFGTX [5:4] RATE = 00b full rate
* (4 bit per clock) 3.125GHz * 4 = 12.5Gbps max speed per lane
* CFGTX [3:1] BUSWIDTH = 10b (Fixed value by HyperLink hardware)
* CFGTX [0] ENTX = 1 (Automatically configured by the HyperLink hardware)
***************************************************************************/
HYPERLINK_SERDES_CFGTX0 = 0x001EE785;
HYPERLINK_SERDES_CFGTX1 = 0x001EE785;
HYPERLINK_SERDES_CFGTX2 = 0x001EE785;
HYPERLINK_SERDES_CFGTX3 = 0x001EE785;

/***************************************************************************
* HyperLink SerDes Control and Status 1 register
* SDCS1[16:23] SLEEP_CNT= 0xFF (recommended value)
* SDCS1[24:31] DISABLE_CNT = 0xFF
**************************************************************************/
SERDES_CONTROL_STATUS1 = 0xFFFF0000;

/***************************************************************************
* HyperLink SerDes PLL Register Configuration
* CFGPLL[12:11] LOOP BANDWIDTH = 00b
* CFGPLL[9] VRANGE = 1 value depends on REFCLK, MPY, and RATE settings
* CFGPLL[8:1] MPY = 00101000b (10x) for 312.50MHz, 00110010b (12.5x) for 250MHz,
* 01010000b (20x) for 156.25MHz. Sets PLL_OUTPUT to max speed (3.125GHz).
* CFGPLL[0] ENPLL = 0(Automatically enabled by HyperLink hardware)
***************************************************************************/
HYPERLINK_SERDES_CFGPLL = 0x0000250;

/* Re-lock device control configuration area */
KICK0 = 0x01234567; //KICK0 key setup
KICK1 = 0x01234567; //KICK1 key setup

/* check for HyperLink SerDes PLL lock */
while(!(HYPERLINK_SERDES_STS & 0x00000001));
```

A.2 KeyStone II Device SerDes Programming

SerDes module information for KeyStone II devices is not provided in this user guide. Please check for availability of the SerDes User Guide for KeyStone II Devices on the device product page.

Index

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information 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.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2013, Texas Instruments Incorporated