

# SerDes Implementation Guide for KeyStone I Devices

*High-Performance and Multicore Processors*

## Contents

1	Introduction .....	4
1.1	Purpose & Scope .....	4
1.2	Abstract .....	4
1.3	Industry Standards Compatibility .....	5
2	General PCB Routing Recommendations .....	6
2.1	Minimum PCB Stack up .....	6
2.2	General Trace/Space and Via Sizes .....	6
2.3	SerDes Interface General Routing Requirements .....	7
2.3.1	Receiver End .....	7
2.3.2	Transmitter End .....	7
2.3.3	Interconnect Guidelines - General .....	7
2.4	Connectors (Optional) .....	8
2.5	Cabling (Optional) .....	8
2.6	Power Supply Requirements .....	8
3	AIF Interface .....	9
3.1	Relevant Industry Standard Specification Support .....	9
3.2	Recommended SerDes PCB Layout Constraints .....	9
3.3	Recommended SerDes Register Configuration Options .....	10
4	SRIO Interface .....	12
4.1	Relevant Industry Standard Specification Support .....	12
4.2	Recommended SerDes PCB Layout Constraints .....	12
4.3	Recommended SerDes Register Configuration Options .....	13
5	HyperLink Interface .....	15
5.1	Relevant Industry Standard Specification Support .....	15
5.2	Recommended SerDes PCB Layout Constraints .....	15
5.3	Recommended SerDes Register Configuration Options .....	16
6	SGMII Interface .....	18
6.1	Relevant Industry Standard Specification Support .....	18
6.2	Recommended SerDes PCB Layout Constraints .....	18
6.3	Recommended SerDes Register Configuration Options .....	19
7	PCIe Interface .....	21
7.1	Relevant Industry Standard Specification Support .....	21
7.2	Recommended SerDes PCB Layout Constraints .....	21
7.3	Recommended SerDes Register Configuration Options .....	22
8	SerDes Configuration common to AIF2, SRIO, and HyperLink .....	24
8.1	SerDes PLL Configuration .....	24
8.1.1	Enabling the PLL .....	24
8.1.2	Reference Clock Multiplication .....	24
8.1.3	VCO Speed Range .....	24
8.1.4	Jitter and PLL Loop Bandwidth .....	25
8.2	SerDes Receiver Configuration .....	25
8.2.1	Data Rate Selection .....	26



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this document.

8.2.2	Receiver Termination .....	26
8.2.3	Adaptive Equalizer .....	26
8.2.3.1	Pre and Post Cursor Equalization Analysis .....	27
8.2.4	Differential Sense Amplifiers .....	28
8.2.5	Clock Recovery Algorithms .....	29
8.2.5.1	First Order Clock Recovery Algorithm .....	30
8.2.5.2	Second Order Clock Recovery Algorithm .....	31
8.2.6	Jitter Tolerance .....	32
8.2.7	Serial to Parallel conversion .....	32
8.2.8	Symbol Alignment .....	32
8.2.8.1	Comma-Based Alignment .....	33
8.2.8.2	Single Bit Alignment Jog .....	34
8.2.9	Loss of Signal Detection .....	34
8.3	SerDes Transmitter Configuration .....	34
8.3.1	Data Rate Selection .....	35
8.3.2	MSYNC .....	35
8.3.3	Serial to Parallel Conversion .....	35
8.3.4	Differential Voltage-Mode Driver .....	35
8.3.5	Output Voltage Swing Control .....	35
8.3.6	Pre and Post Cursor FIR filter .....	35
9	SerDes Configuration Common to PCIe and SGMII .....	38
9.1	SerDes PLL Configuration .....	38
9.1.1	Enabling The PLL .....	38
9.1.2	Reference Clock Multiplication .....	38
9.1.3	VCO Speed Range .....	38
9.1.4	Jitter and PLL Loop Bandwidth .....	38
9.2	SerDes Receiver Configuration .....	39
9.2.1	Data Rate Selection .....	40
9.2.2	Receiver Termination .....	40
9.2.3	Adaptive Equalizer .....	40
9.2.4	Differential Sense Amplifiers .....	42
9.2.5	Clock Recovery Algorithms .....	42
9.2.5.1	First Order Clock Recovery Algorithm .....	43
9.2.5.2	Second Order Clock Recovery Algorithm .....	44
9.2.6	Jitter Tolerance .....	45
9.2.7	Serial to Parallel conversion .....	46
9.2.8	Symbol Alignment .....	46
9.2.8.1	Comma-Based Alignment .....	46
9.2.8.2	Single Bit Alignment Jog .....	47
9.2.9	Loss of Signal Detection .....	47
9.3	SerDes Transmitter Configuration .....	48
9.3.1	Data Rate Selection .....	48
9.3.2	Serial to Parallel Conversion .....	48
9.3.3	Differential Voltage-Mode Driver .....	48
9.3.4	Output Voltage Swing Control .....	49
9.3.5	Output Common Mode Adjustment .....	49
9.3.6	De-emphasis .....	49
10	SerDes Lane-Setting Optimization Procedures .....	51
10.1	Optimizing PLL Performance for Reducing TX Jitter and Increasing RX CDR Margin .....	51
10.2	Optimizing Transmitter SWING and FIR Filter Settings for Optimal Power and Receiver Performance .....	51
10.2.1	Creating Initial Values for Swing and FIR Settings .....	52
10.2.2	Creating Valid Swing and FIR Settings Map Based on Receiver Errors Detected .....	52
11	Terminations .....	54
12	Related Documentation From Texas Instruments .....	55
13	Revision History .....	56

## List of Tables

Table 1	Minimum PCB Stack Up	6
Table 2	CPRI Physical Layer Modes	9
Table 3	PLL Loop Bandwidth Selection (MHz)	25
Table 4	Receiver Operating Rate	26
Table 5	Receiver Equalizer Configuration	26
Table 6	Receiver Equalizer Hold	27
Table 7	Receiver Polarity	28
Table 8	Offset Compensation	29
Table 9	Clock Data Recovery Algorithm selection	30
Table 10	Bus Width	32
Table 11	Receiver Symbol Alignment Selection	32
Table 12	Precursor Transmit Tap Weights	36
Table 13	Transmit Postcursor 1 Tap Weights (Nominal)	37
Table 14	PLL Loop Bandwidth Selection (MHz)	39
Table 15	Receiver Operating Rate	40
Table 16	Receiver Equalizer Configuration (EQ)	41
Table 17	Clock Data Recovery Algorithm Selection	43
Table 18	Second Order Clock Recovery Algorithm Parameters	45
Table 19	Receiver Symbol Alignment Selection	46
Table 20	Receiver Operating Rate	48
Table 21	Differential Output Swing	49
Table 22	Differential Output De-emphasis	49

## List of Figures

Figure 1	Receiver Data and Phase Samples	29
Figure 2	rcbclk[i] and rdi[19:0] Interface	33
Figure 3	Transmitter FIR Block Diagram	36
Figure 4	Receiver Data and Phase Samples	42
Figure 5	rcbclk[i] and rdi[19:0] Interface	46

## 1 Introduction

### 1.1 Purpose & Scope

The goal of KeyStone I SerDes collateral material is to make system implementation easier for the customer by providing the system solution. For these SerDes-based interfaces, it is not assumed that the system designer is familiar with the industry specifications, SerDes technology, or RF/microwave PCB design. However, it is still expected that the PCB design work will be supervised by a knowledgeable high-speed digital PCB designer and an assumption is made that the PCB designer is using established high-speed design rules.

This document is intended to aid in the hardware design and implementation of a KeyStone I-based system. The document should be used along with the device-specific data manual and relevant user guides, application reports, standards, and specifications (see [“Related Documentation From Texas Instruments”](#) on page 55).

### 1.2 Abstract

This document contains implementation instructions for the serializer/deserializer (SerDes)-based interfaces on the KeyStone I family of DSP devices. These include:

- Serial RapidIO® (SRIO)
- Antenna Interface (AIF)
- HyperLink.
- Serial Gigabit Media Independent Interface (SGMII) interfaces
- Peripheral Component Interconnect Express (PCIe)

Serial RapidIO is an industry-standard high-speed switched-packet interconnects. The Antenna Interface is compatible with two industry standards targeted at cellular base station solutions; Open Base Station Architecture Initiative (OBSAI) and Common Public Radio Interface (CPRI). SGMII is a standard used for gigabit Ethernet connections from MAC to MAC or MAC to PHY. Peripheral Component Interconnect Express (PCIe) is a high speed serial interconnect. HyperLink is a Texas Instruments high-speed packet-based interconnect.

For each of these interfaces, physical layer data transmission uses analog SerDes to feed low-output-swing differential current-mode logic (CML) buffers. Proper printed circuit board (PCB) design for these interfaces resembles analog or RF design, and is very different than traditional parallel digital bus design.

Due to this analog nature of SerDes-based interfaces, it is not possible to specify the interface in a traditional DSP digital interface manner. Furthermore, it is undesirable to specify the interface in terms of the raw physical requirements laid out by the industry standard specifications. Understanding these specifications and producing a compliant PCB-based on the explicit and implicit requirements there demands significant time, experience, and expensive tools.

For KeyStone I SerDes-based interfaces, the approach is to reduce the specifications to a set of easy-to-follow PCB routing rules and system configurations. TI has performed the simulation and system design work to ensure the appropriate interface requirements are met. This document describes guidelines that, when followed, result in board level implementations that meet the interface requirements.

### 1.3 Industry Standards Compatibility

All SerDes interfaces are configured as point-to-point connections. It is assumed that the connection is made between a KeyStone I SoC and another device compliant to the appropriate industry standard. The list of supported standards is given below. Note that this document deals with the physical layer and, therefore, it is the electrical specifications in these standards that are relevant. For more information regarding protocol compliance<sup>1</sup>, see the device-specific user guides in “[Related Documentation From Texas Instruments](#)” on page 55.

- **Serial RapidIO:** This is electrically compliant with Serial RapidIO specification revision 2.0
- **Antenna Interface:** This support both OBSAI and CIPRI interfaces
  - **OBSAI** interface is electrically compliant with the OBSAI RP3 specification version 4.1; RP1 specification version 2.1
  - **CPRI** interface is electrically compliant with the low voltage variant of the CPRI version 4.1 specification (guided by XAUI 802.3ae Clause 47).
- **SGMII:** This is electrically compliant with SGMII revision 1.8 with the following clarifications
  - It does not implement the separate clock signaling
  - It must be AC coupled and may require external terminations
  - Electrical compatibility does not guarantee interoperability with devices.
- **Peripheral Component Interconnect Express:** This electrically compliant with version 2.1.
- **HyperLink:** This is a Texas Instruments Interface that provides a high-speed, low-latency, and low-pin-count communication interface between two KeyStone I SoC devices. For more details, see the *HyperLink Users Guide* in “[Related Documentation From Texas Instruments](#)” on page 55.

<sup>1</sup>.Electrical compatibility does not guarantee interoperability with devices

## 2 General PCB Routing Recommendations

### 2.1 Minimum PCB Stack up

The minimum PCB stack up for routing the KeyStone I devices is considered to be a six-layer stack up as described in [Table 1](#). This assumes minimal peripherals are used. Combination of peripherals will increase PCB stack up complexity and layer count. It is feasibly possible to route all peripherals in a 12-14 layer board if priority is given to high performance interfaces, otherwise your layer count (including power planes) can grow to as many as 18 layers.

**Table 1 Minimum PCB Stack Up**

Layer	Type	Description
1	Signal	Top routing
2	Plane	Ground
3	Plane	Split power
4	Signal	Internal routing
5	Plane	Ground
6	Signal	Bottom routing
<b>End of Table 1</b>		

Additional layers may be added as needed. All layers with SerDes traces must be able to achieve 100- $\Omega$  differential impedance.

### 2.2 General Trace/Space and Via Sizes

The key concern for SerDes signal traces is the need to achieve 100- $\Omega$  differential impedance. This differential impedance is impacted by trace width, trace spacing, distance between planes, and dielectric material. Verify with a proper PCB manufacturing tool that the trace geometry for all SerDes traces results in exactly 100- $\Omega$  differential impedance traces.

Of secondary concern is the insertion loss caused by the traces. Due to the skin effect, wider traces have lower losses than narrower traces. Therefore, longer SerDes runs should use wider traces for lower loss. However, be aware that layers in the stack up that are set to 100- $\Omega$  differential impedance with wider traces may be less desirable for routing other signals. Trace widths are recommended to be 5 mil (0.127 mm) maximum, typical SerDes Applications commonly use 4 mil (0.1 mm) width traces for maximum lengths of up to 10 inches.

Standard via sizes that allow escape from a 0.8-mm pitch device can be used (i.e., 8-mil holes, 18-mil pads). Micro and/or blind/buried vias are neither required nor prohibited.

The PCB BGA pad requirements for the KeyStone I devices are documented in the *Flip Chip Ball Grid Array Package Reference Guide* (SPRU811), in [“Related Documentation From Texas Instruments”](#) on page 55. Most current KeyStone I DSPs are designed around a 0.8-mm ball pitch package and should follow the 0.8-mm guidelines. The PCB BGA pad requirements for the SerDes link partner device should follow its manufacturer's guidelines.

## 2.3 SerDes Interface General Routing Requirements

The approach for specifying suitable SerDes routing breaks the physical connection down into three component pieces: receiver end, transmitter end, and interconnect. The receiver and transmitter end are the pieces closest to the packages of the connected devices. The receiver end is connected to the SerDes input of the device and goes from the BGA pads to the AC-coupling capacitors. The transmitter end is connected to the SerDes output of the device and is simply the BGA escape paths for the differential pairs. The interconnect joins the receiver and transmitter ends.

### 2.3.1 Receiver End

For the receiver end, it is strongly recommended to route the trace from the BGA pad to the capacitor pad on the top layer. This avoids a via escape between the BGA pad and the capacitor. On the other side of the capacitor, it is recommended to via to another layer. The trace widths and separation should be altered based on the board stack up to meet the 100- $\Omega$  differential impedance requirement. Traces may be necked down to escape the BGA, if necessary.

### 2.3.2 Transmitter End

The transmitter end should use standard via escapes to internal layers. Internal layers are recommended for their superior shielding characteristics. The trace widths and separation should be selected based on the board stack up to meet the 100- $\Omega$  differential impedance requirement. Traces may be necked down to escape the BGA, if necessary.

### 2.3.3 Interconnect Guidelines - General

The geometry of the traces to link the transmitter and receiver ends is determined by the placement in the target system and any board-to-board connections. The trace can be placed as required, as long as it meets the following requirements:

- Edge-coupled, matched-length differential pair
- No stubs
- Maximum trace lengths (simulation and modeling recommended to guarantee performance goals are met, skew and net classes per the *KeyStone I Hardware Design Guide*, see [“Related Documentation From Texas Instruments”](#) on page 55.)
- 100- $\Omega$  differential impedance
- The areas where desired differential pair separation cannot be maintained (connections to devices or connectors) should be kept to an absolute minimum
- Do not route across split planes in the neighboring reference plane
- Maintain uniform separation between complementary pairs for the entire trace length
- No more than two sets of vias (not including via for BGA breakouts) where possible, modeling is strongly recommended when three or more sets of vias are used)
- Whenever possible, use the majority of via length to transfer signal layers in order to avoid via stubs (back drill vias where possible to remove via stubs)
- Other signals are separated by at least 2 $\times$  the differential spacing
- Internal layers are strongly preferred. Avoid top and bottom layers (except for escapes)
- SerDes routing should be adjacent to a reference plane (either ground or power) and within one signal plane of a ground reference plane

- If connectors are used, they must be of a suitable 100- $\Omega$  differential-impedance, high-speed type, and count as a minimum of 1" of trace for each connector pair (simulation and modeling is strongly recommended)
- If a mid bus probe is used (not recommended), it must follow both TI's and the probe manufacturer's recommended guidelines, and shall count as 2" of additional trace (plus parasitic loading).

## 2.4 Connectors (Optional)

Any connectors used must be controlled impedance (50- $\Omega$  single ended or 100- $\Omega$  differential) and suitable for microwave transmissions. Suitable connectors are typically categorized as backplane type connectors. The connectors should have less than 1-dB insertion loss below 6 GHz. Some suggested connectors are:

- CN074 - AMC Connector
- Tyco Z-DO
- Tyco Z-PAK HM Z

## 2.5 Cabling (Optional)

Any cabling used must be controlled impedance (50- $\Omega$  single ended or 100- $\Omega$  differential) and suitable for microwave transmissions. Recommended cable types are listed below:

- 50- $\Omega$  Coaxial - commonly used with SMA connectors
  - RG142
  - RG316
  - RG178
- Infiniband - assembled cables available in 1 $\times$  and 4 $\times$  widths

## 2.6 Power Supply Requirements

The power supply and bypassing requirements for SerDes power planes are documented as part of the *KeyStone I Hardware Design Guide*, see [“Related Documentation From Texas Instruments”](#) on page 55.

It is best to use power plane splits to connect the power supply from the filters to the pins. However, traces that are at least 20 mils wide can also be used to access the inner BGA pads.

## 3 AIF Interface

The enhanced Antenna Interface subsystem (AIF2) on KeyStone I devices consists of the Antenna Interface module and two SerDes macros. This is a six-lane SerDes interface designed to operate at up to 6.144 Gbps per lane from pin-to-pin. The AIF2 relies on the performance SerDes macro along with a logic layer for the OBSAI RP3 and CPRI protocols. The AIF is used to connect to the backplane for transmission and reception of antenna data, as well as to connect to additional device peripherals.

For more details on the AIF interface, see the device-specific data manuals. Specifications supported, interface specific PCB routing recommendations, SerDes device settings accessible by register accesses are covered in further sections of this document.

### 3.1 Relevant Industry Standard Specification Support

The parameters for the AC electrical specifications for OBSAI support are guided by existing standards. For the line rates up to 3072 MBaud, the XAUI electrical interface specified in Clause 47 of IEEE 802.3ae-2002 is used. For the 6144 MBaud line rate, the references are OIF-CEI-02.0 Interoperability Agreement with its section 7 and related clauses and the Serial RapidIO v2 PHY specifications, which are also based on the OIF agreement, to be adapted to the specific needs in OBSAI. OBSAI support includes the interconnects that are in compliance with TYPE 1, TYPE 2, TYPE 3, TYPE 4 and TYPE 5 interconnects as specified in Chapter 5 of OBSAI RP3 Rev 4.2 Specification.

Two electrical variants of CPRI supported are LV:XAUI-based and LV-II:CEI-6G-LR-based. The LV variant is guided by IEEE 802.3-2005 clause 47 (XAUI), but with a lower bit rate. The LV-II variant is guided by OIF-CEI-02.0, clause 7, but with a lower bit rate. [Table 2](#) summarizes the line rates and electrical standards supported on the CPRI interface as specified in Chapter 4 of the CPRI Interface Specification V5.0.

**Table 2 CPRI Physical Layer Modes**

Line Bit Rate	Electrical Std.	LV:XAUI	LV-II:CEI-6G-LR
614.4 Mbit/s	E.6	Supported	Supported
1228.8 Mbit/s	E.12	Supported	Supported
2457.6 Mbit/s	E.24	Supported	Supported
3072.0 Mbit/s	E.30	Supported	Supported
4915.2 Mbit/s	E.48	Not applicable	Supported
6144.0 Mbit/s	E.60	Not applicable	Supported

### 3.2 Recommended SerDes PCB Layout Constraints

Routing requirements for the AIF (RP3) interface must adhere to good engineering practices for transmission lines operating at or above 1 GHz. Specific attention must be paid to net classes within this group and should have a high routing priority. The device incorporates SerDes outputs, which typically connect directly to the antenna interface (through a back plane) or between devices (for serial and parallel processing).

- Each complementary device SerDes receive pair must be individually skew-matched to within 5 ps. 5 ps equates to approximately 27.32 mils to 35.46 mils (depending on propagation delays). Example of complementary pairs include AIFRXN0 & AIFRXP0.
- Both complementary device SerDes receive pairs must be routed on the same layer.

- Each complementary device SerDes transmit pairs must be skew-matched to within 5 ps. 5 ps equates to approximately 27.32 mils to 35.46 mils (depending on propagation delays). Example of complementary SerDes pairs include: AIFTXN0 & AIFTXP0.
- All complementary device transmit pairs must be routed on the same layer.
- All complementary device receive pairs (AIFRXN/P5:0) must be assigned to an individual net class and routing skew must not be greater than 10 ps between all receive pairs.
- All complementary device transmit pairs (AIFTXN/P5:0) must be assigned to an individual net class and routing skew must not be greater than 10 ps between all transmit pairs.
- Transmit and receive signals must be referenced to parallel ground planes.
- Vias are allowed and should never exceed two per net, all nets must be balanced and the impact of the via on timing and loading taken into account during design and layout.
- All net signals must be referenced to a parallel ground plane.
- Differential signal routing must achieve a 100- $\Omega$  differential impedance.

### 3.3 Recommended SerDes Register Configuration Options

The AIF interface is made up of 2 macro instances B8 and B4. B8 is used to support 8 lanes (4 pairs of RX+TX) and B4 is used to support 4 lanes (2 pairs of RX+TX).

The PLL configuration within the AIF SerDes interface is configured using:

- The AIF SerDes PLL Configuration Register (SD\_PLL\_B8\_EN\_CFG, SD\_PLL\_B4\_EN\_CFG, SD\_PLL\_B8\_CFG, SD\_PLL\_B4\_CFG)
- The AIF SerDes PLL Status Register (SD\_PLL\_B8\_STS, SD\_PLL\_B4\_STS).

The Receiver configuration within the AIF SerDes interface is configured using:

- The AIF SerDes Configuration Register (SD\_RX\_EN\_CFG, SD\_RX\_R1\_CFG, SD\_RX\_R2\_CFG)
- The AIF Receiver SerDes Status Register (SD\_RX\_STS) specific to a lane.

The Transmitter configuration within the AIF SerDes interface is configured using:

- The AIF SerDes Configuration Register (SD\_TX\_EN\_CFG, SD\_TX\_R1\_CFG, SD\_TX\_R2\_CFG)
- The AIF Transmitter SerDes Status Register (SD\_TX\_STS) specific to a lane.

For details on AIF SerDes configuration registers definitions and settings, see the *KeyStone Architecture Antenna Interface 2 (AIF2) User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

An example SerDes Configuration for AIF interface is shown below. For the complete suite of configuration examples available from TI, see the Multicore Software Development Kit (MCSDK).

### Example 1 AIF SerDes Configuration

```
-----
//SD common setup

SdCommonSetup.bEnablePllB8 = TRUE;
SdCommonSetup.CLKBYP_B8 = CSL_AIF2_PLL_CLOCK_NO_BYPASS;
SdCommonSetup.LB_B8 = CSL_AIF2_PLL_LOOP_BAND_MID; //High BW is also fine
SdCommonSetup.VoltRangeB8 = CSL_AIF2_PLL_VOLTAGE_LOW; //fixed factor
SdCommonSetup.SleepPllB8 = CSL_AIF2_PLL_AWAKE;
SdCommonSetup.pllMpyFactorB8 = CSL_AIF2_PLL_MUL_FACTOR_25X; //for OBSAI when
reference clock is 122.88 Mhz
SdCommonSetup.SysClockSelect = CSL_AIF2_SD_BYTECLOCK_FROM_B8;
SdCommonSetup.DisableLinkClock[0] = FALSE; //enable link0 clock

//SD link setup
SdLinkSetup.rxAlign = CSL_AIF2_SD_RX_COMMA_ALIGNMENT_ENABLE; //01 Comma alignment
enabled
SdLinkSetup.rxLos = CSL_AIF2_SD_RX_LOS_ENABLE; //100 Enabled
SdLinkSetup.rxCdrAlgorithm = CSL_AIF2_SD_RX_CDR_FIRST_ORDER_THRESH_1; //000 First
order, threshold of 1 (Suitable for asynchronous
system with Low frequency offset)
SdLinkSetup.rxInvertPolarity = CSL_AIF2_SD_RX_NORMAL_POLARITY; //0 Normal
polarity
SdLinkSetup.rxTermination = CSL_AIF2_SD_RX_TERM_COMMON_POINT_0_7 ; //001 Common
point set to 0.7VDDT for AC coupled application
SdLinkSetup.rxEqualizerConfig = CSL_AIF2_SD_RX_EQ_ADAPTIVE; //001 Fully adaptive
equalization (EQ On)
SdLinkSetup.brxEqHold = FALSE; //0 EQ adaptation enabled (fixed value)
SdLinkSetup.brXOffsetComp = TRUE; //1 Enabled (fixed value)
SdLinkSetup.bEnableTxSyncMater = TRUE; //1 Enable (fixed value)
SdLinkSetup.txInvertPolarity = CSL_AIF2_SD_TX_PAIR_NORMAL_POLARITY; //0 Normal
polarity
SdLinkSetup.txOutputSwing = CSL_AIF2_SD_TX_OUTPUT_SWING_14; //1110 1200mv
SdLinkSetup.txPrecursorTapWeight = CSL_AIF2_SD_TX_PRE_TAP_WEIGHT_2; //010 -5.0%
SdLinkSetup.txPostcursorTapWeight = CSL_AIF2_SD_TX_POST_TAP_WEIGHT_24; //10000
-20%
SdLinkSetup.bTxFirFilterUpdate = TRUE; //FIR filter update on
```

### End of Example 1

## 4 SRIO Interface

The SRIO port on the KeyStone I device is a high-performance, low pin-count SerDes interconnect. This is a four-lane SerDes interface designed to operate up to 5 GBaud per lane from pin-to-pin. RapidIO is based on the memory and device addressing concepts of processor buses in which the transaction processing is managed completely by hardware.

For more details on SRIO interface, see the device-specific data manuals. Specifications supported, interface specific PCB routing recommendations, and SerDes device settings accessible by register accesses are covered in further sections of this document.

### 4.1 Relevant Industry Standard Specification Support

The LP-Serial 1.25 Gbaud, 2.5 Gbaud, and 3.125 Gbaud Electrical specifications are guided by the XAUI electrical interface specified in Clause 47 of IEEE 802.3ae-2002. The LP-Serial 5 Gbaud Electrical specifications are based upon the Optical Internetworking Forum's Common Electrical Interface CEI. SRIO interface shall support 1.25 Gbaud, 2.5 Gbaud, or 3.125 Gbaud baud rates on Level I link and 5 Gbaud baud rates on Level II links as described in Chapter 8 of RapidIO Interconnect Specification Part 6: LP-Serial Physical Layer Specification Rev. 2.1.

### 4.2 Recommended SerDes PCB Layout Constraints

Routing requirements for the SRIO interface to the device shall adhere to good engineering practices for transmission lines operating above 6.25 GHz (actual operating frequency is 5 GHz). Specific attention shall be paid to net classes within this group and should have a high routing priority.

- Each complementary SRIO SerDes receive pair shall be individually skew matched to within 1 ps. 1 ps equates to approximately 5.464 mils to 7.092 mils (depending on propagation delays). Example of the SRIO complementary pairs include RIORXN0 & RIORXP0.
- All four complementary SRIO SerDes receive pairs shall be routed on the same layer.
- All four complementary transmit pairs shall be routed on the same layer.
- All four complementary receive pairs RIORXN/P3:0 shall be assigned and routed as an individual net class, routing skew shall not be greater than 15 ps between all receive pairs.
- All 4 complementary transmit pairs RIOTXN/P3:0 shall be assigned and routed as an individual net class, routing skew shall not be greater than 15 ps between all transmit pairs.
- Transmit and receive signals must be referenced to parallel ground planes.
- Vias are allowed and should never exceed 2 per complete net, all nets must be balanced and the impact of the via on timing and loading taken into account during design and layout.
- Differential signal routing must achieve a 100- $\Omega$  differential impedance.
- If a SRIO switch is used, the specific routing and timing requirements shall also be incorporated.

### 4.3 Recommended SerDes Register Configuration Options

The PLL configuration within the SRIo SerDes interface is configured using this register:

- SRIo SerDes PLL Configuration Register (SRIo\_SERDES\_CFGPLL)

The Receiver configuration within the SRIo SerDes interface is configured using this register:

- SRIo SerDes Configuration Register (SRIo\_SERDES\_CFGRXn)

The Transmitter configuration within the SRIo SerDes interface is configured using this register:

- SRIo SerDes Configuration Register (SRIo\_SERDES\_CFGTXn)

For details on SRIo SerDes configuration registers definitions and settings, see the *KeyStone Architecture SRIo User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

An example SerDes Configuration for SRIo interface is shown below. For the complete suite of configuration examples available from TI, see the Multicore Software Development Kit (MCSDK).

#### Example 2 SRIo SerDes Configuration

```

-----
/* Set rx/tx config values based on the lane rate specified */
switch (linkRateGbps)
{
    case srio_lane_rate_5p000Gbps:/* Pll setting determines 5.0 Gbps or 3.125
Gbps */
    case srio_lane_rate_3p125Gbps:/* Same Tx and Rx settings for 5.0 Gbps or
3.125 Gbps */
        rxConfig = 0x00440495;
        // (0) Enable Receiver
        // (1-3) Bus Width 010b (20 bit)
        // (4-5) Half rate. Two data samples per PLL output clock cycle
        // (6) Normal polarity
        // (7-9) Termination programmed to be 001
        // (10-11) Comma Alignment enabled
        // (12-14) Loss of signal detection disabled
        // (15-17) First order. Phase offset tracking up to +-488 ppm
        // (18-20) Fully adaptive equalization
        // (22) Offset compensation enabled
        // (23-24) Loopback disabled
        // (25-27) Test pattern mode disabled
        // (28-31) Reserved
        txConfig = 0x00180795;
        // (0) Enable Transmitter
        // (1-3) Bus Width 010b (20 bit)
        // (4-5) Half rate. Two data samples per PLL output clock cycle
        // (6) Normal polarity
        // (7-10) Swing max.
        // (11-13) Precursor Tap weight 0%
        // (14-18) Adjacent post cursor Tap weight 0%
        // (19) Transmitter pre and post cursor FIR filter update
        // (20) Synchronization master
        // (21-22) Loopback disabled
        // (23-25) Test pattern mode disabled
        // (26-31) Reserved
        break;
    case srio_lane_rate_2p500Gbps:/* Tx and Rx settings for 2.50 Gbps */
        rxConfig = 0x004404A5;
        // (4-5) Quarter rate. One data sample per PLL output clock cycle
        txConfig = 0x001807A5;
        // (4-5) Quarter rate. One data sample per PLL output clock cycle
        break;
    case srio_lane_rate_1p250Gbps:/* Tx and Rx settings for 1.25 Gbps */
        rxConfig = 0x004404B5;
        // (4-5) Eighth rate. One data sample every two PLL output clock cycles
        txConfig = 0x001807B5;
        // (4-5) Eighth rate. One data sample every two PLL output clock cycles
}

```

```
        break;
    default:/* Invalid SRIO lane rate specified */
        return -1;
}
```

**End of Example 2**

-----

## 5 HyperLink Interface

The KeyStone I devices includes the HyperLink for companion chip/die interfaces. This is a four-lane SerDes interface designed to operate up to 12.5 Gbps per lane from pin-to-pin. The interface is used to connect with external accelerators that are manufactured using TI libraries. 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.

For more details on HyperLink interface, see the device specific-data manuals. Specifications supported, interface specific PCB routing recommendations, SerDes device settings accessible by register accesses and are covered in further sections of this document.

### 5.1 Relevant Industry Standard Specification Support

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

### 5.2 Recommended SerDes PCB Layout Constraints

- Each differential receive pair shall be individually skew matched to within 1 ps (absolute maximum). 1 ps equates to approximately 5.5 mils to 7.1 mils (depending on propagation delays). An example of a differential pair is MCMRXN0 & MCMRXP0.
- All differential receive pairs shall be routed on the same layer.
- Each differential transmit pair shall be individually skew matched to within 1 ps (absolute maximum). 1 ps equates to approximately 5.5 mils to 7.1 mils (depending on propagation delays). An example of a differential pair is MCMTXN0 & MCMTXP0.
- All differential transmit pairs shall be routed on the same layer.
- All differential receive pairs MCMRXN/P3:0 shall be assigned to an individual net class and routing skew shall not be greater than 100 ps (absolute maximum) between all receive pairs.
- All differential transmit pairs MCMTXN/P3:0 shall be assigned to an individual net class and routing skew shall not be greater than 100 ps (absolute maximum) between all transmit pairs.
- All differential pairs shall be < 4.0" (101.6 mm) in total length, recommended length is 2.00" (50.8 mm).
- The MCMRXFLCLK & MCMRXFLDAT nets shall be skew matched within 250 ps (absolute maximum) of one another.
- The MCMTXFLCLK & MCMTXFLDAT nets shall be skew matched within 250 ps (absolute maximum) of one another.
- The MCMRXPMCLK & MCMRXPMDAT nets shall be skew matched within 250 ps (absolute maximum) of one another.
- The MCMTXPMCLK & MCMTXPMDAT nets shall be skew matched within 250 ps (absolute maximum) of one another.
- Transmit and receive signals must be referenced to continuous, parallel ground planes.
- Differential signal routing must achieve 100-ohm differential impedance.

- Routing shall take into account propagation delays between microstrip and strip line topologies.
- To prevent crosstalk in a simple board stack-up, we recommend that the differential receive pairs be routed as microstrip (outer layer) on one side of the board and the differential transmit pairs be routed as microstrip (outer layer) on the other side of the board.
- Up to 2 vias are allowed but there must be no stub and the differential nature of the circuit must be maintained. This forces routing on the outer layers unless blind via technologies or back-drilling are used.
- If a via is implemented, an adjacent ground via will be implemented to minimize the discontinuity.
- HyperLink lanes can be swapped to simplify routing. The differential pairing must be maintained.
- P and N connections for a single differential pair can be inverted to simplify routing.
- The HyperLink interface is intended for DC coupled operation between two DSPs on a single board.

Board layout simulation is a requirement for this class of circuit to validate the PCB routing.

### 5.3 Recommended SerDes Register Configuration Options

The PLL configuration within the HyperLink SerDes interface is configured using:

- The HyperLink SerDes PLL Configuration Register (HYPERLINK\_SERDES\_CFGPLL)

The Receiver configuration within the HyperLink SerDes interface is configured using:

- The HyperLink SerDes Configuration Register (HYPERLINK\_SERDES\_CFGRXn)
- The HyperLink Receiver SerDes Status Register (HYPERLINK\_SERDES\_STS).

The Transmitter configuration within the HyperLink SerDes interface is configured using:

- The HyperLink SerDes Configuration Register (HYPERLINK\_SERDES\_CFGTXn)
- The HyperLink Transmitter SerDes Status Register (HYPERLINK\_SERDES\_STS)

For details on HyperLink SerDes configuration registers definitions and settings, see the *KeyStone Architecture HyperLink User Guide*, in “[Related Documentation From Texas Instruments](#)” on page 55.

An example SerDes Configuration for HyperLink interface is shown below. For the complete suite of configuration examples available from TI, see the Multicore Software Development Kit (MCSDK).

### Example 3 HyperLink SerDes Configuration

```

-----
//unlock device control configuration area to setup HyperLink SERDES (TI internal
registers)
mcm_reg_wr((volatile int *) (btcfg_cfg_base+0x38),0x83e70b13); //KICK0 key setup
mcm_reg_wr((volatile int *) (btcfg_cfg_base+0x3c),0x95a4f1e0); //KICK1 key setup
/** HyperLink SERDES configuration (User may use SRIO SERDES register map for
this)*****/

/* Configure CFGRX
CFGRX [0] ENRX = 1 Don't care (Overwritten automatically by HyperLink)  CFGRX [3:1]
BUSWIDTH = 2
CFGRX [5:4] RATE = 0 full rate (4 bit per clock) 3.125GHz * 4 = 12.5Gbps max speed
per lane
CFGRX [6] INVPAIR = 0  CFGRX [9:7] TERM = 1 (1 for AC couple and 6 for DC couple)
CFGRX [11:10] ALIGN = 01 Don't care (Overwritten by HyperLink in run time)
CFGRX [14:12] LOS = 100 (VUSR automatically sets to 0 for loopback. User must set
to 4 for non-loopback)
CFGRX [17:15] CDR = 101          CFGRX [20:18] EQ = 001
CFGRX [21] EQHLD = 0          CFGRX [22] ENOC = 1
CFGRX [24:23] LOOPBACK = 00 Automatically configured by HyperLink HW
CFGRX [27:25] TESTPATTERN = 000 */
mcm_reg_wr((volatile int *) (btcfg_cfg_base+VUSR_SERDES_CFGRX0), 0x0046C485);
mcm_reg_wr((volatile int *) (btcfg_cfg_base+VUSR_SERDES_CFGRX1), 0x0046C485);
mcm_reg_wr((volatile int *) (btcfg_cfg_base+VUSR_SERDES_CFGRX2), 0x0046C485);
mcm_reg_wr((volatile int *) (btcfg_cfg_base+VUSR_SERDES_CFGRX3), 0x0046C485);

/* Configure CFGTX
CFGTX [0] ENTX = 1 Don't care (Overwritten automatically by HyperLink)  CFGTX
[3:1] BUSWIDTH = 2
CFGTX [5:4] RATE = 0 full rate (4 bit per clock) 3.125GHz * 4 = 12.5Gbps max speed
per lane
CFGTX [6] INVPAIR = 0
CFGTX [10:7] SWING = 1110b 7 for short trace length and 15 for long trace length
CFGTX [13:11] TWPRE = 001b Precursor Tap weight. (-2.5%)
CFGTX [18:14] TWPST1 = 18(10010b) Adjacent Post Cursor Tap Weight
If trace length is 4" or less, use 23 (-10%). If trace length is between 4" and
10", use 27 (-27.5%).
CFGTX [19] FIRUPT = 1  Transmitter pre and post cursor FIR filter update
CFGTX [20] MSYNC = 1 (HyperLink will automatically set this value)
CFGTX [22:21] LOOPBACK = 00 (HyperLink will automatically set this field)
CFGTX [25:23] TESTPATTERN = 000 */
mcm_reg_wr((volatile int *) (btcfg_cfg_base+VUSR_SERDES_CFGTX0), 0x001C8F05 );
mcm_reg_wr((volatile int *) (btcfg_cfg_base+VUSR_SERDES_CFGTX1), 0x001C8F05 );
mcm_reg_wr((volatile int *) (btcfg_cfg_base+VUSR_SERDES_CFGTX2), 0x001C8F05 );
mcm_reg_wr((volatile int *) (btcfg_cfg_base+VUSR_SERDES_CFGTX3), 0x001C8F05 );

/* Configure SERDES PLL
CFGPLL[0] ENPLL = 0 (HyperLink overrides this value)
CFGPLL[8:1] MPY = 40(10x)for 312.50MHz input clock, 80(20x)for 156.25MHz,
50(12.5x) for 250MHz to get max speed 3.125GHz
CFGPLL[9] VRANGE = 1 for full rate and half rate. 0 for quad rate and eighth rate
CFGPLL[10] SLEEPPLL = 0
CFGPLL[12:11] LOOP BANDWIDTH = 0  CFGPLL[14:13] sCLKBYP = 00 no Bypass */
mcm_reg_wr((volatile int *) (btcfg_cfg_base+VUSR_SERDES_CFGPLL), 0x250);

```

End of Example 3

## 6 SGMII Interface

The gigabit Ethernet (GbE) switch subsystem on KeyStone I devices provides an efficient SGMII interface between the TI SoC and the networked community. The EMAC supports 10Base-T (10 Mbits/second [Mbps]), and 100BaseTX (100 Mbps), in half- or full-duplex mode, and 1000BaseT (1000 Mbps) in full-duplex mode, with hardware flow control and quality-of-service (QOS) support. The GbE switch subsystem is coupled with the network coprocessor.

For more details on GbE interface, see the device-specific data manuals. Specifications supported, interface specific PCB routing recommendations, and SerDes device settings, accessible by register accesses are covered in further sections of this document.

### 6.1 Relevant Industry Standard Specification Support

The SGMII interface adheres with IEEE Standard for low-voltage differential signals (LVDS) for Scalable Coherent Interface (SCI) IEEE1596.3-1996.

### 6.2 Recommended SerDes PCB Layout Constraints

Routing requirements for the SGMII or Ethernet interface must adhere to good engineering practices for transmission lines operating at or above 1 GHz. Specific attention must be paid to net classes within this group and should have a high routing priority. The device incorporates SerDes outputs and requires the use of a PHY to interconnect to a standard RJ-45 connector.

- Each complementary device SerDes receive pair must be individually skew-matched to within 5 ps. 5 ps equates to approximately 27.32 mils to 35.46 mils (depending on propagation delays). Example of complementary pairs include SGMIIRXN0 & SGMIIRXP0.
- Both complementary device SerDes receive pairs must be routed on the same layer.
- Each complementary device SerDes transmit pairs must be skew-matched to within 5 ps. 5 ps equates to approximately 27.32 mils to 35.46 mils (depending on propagation delays). Example of complementary SerDes pairs include: SGMII TXN0 & SGMII TXP0.
- All complementary device transmit pairs must be routed on the same layer.
- All complementary device receive pairs (SGMIIRXN/P3:0) must be assigned to an individual net class and routing skew must not be greater than 10 ps between all receive pairs.
- All complementary device transmit pairs (SGMIITXN/P3:0) must be assigned to an individual net class and routing skew must not be greater than 10 ps between all transmit pairs.
- Transmit and receive signals must be referenced to parallel ground planes.
- Vias are allowed and should never exceed two per net, all nets must be balanced and the impact of the via on timing and loading taken into account during design and layout.
- All net signals must be referenced to a parallel ground plane.
- Differential signal routing must achieve a 100-Ω differential impedance.

### 6.3 Recommended SerDes Register Configuration Options

The SerDes macro within SGMII is configured by programming SerDes registers internal to the macro.

The PLL configuration within the SGMII SerDes interface is configured using this register:

- SGMII PLL Configuration Register (SGMII\_SERDES\_CFGPLL)

The Receiver configuration within the SGMII SerDes interface is configured using these registers:

- SGMII Receiver Configuration Registers (SGMII\_SERDES\_CFGRX0 and SGMII\_SERDES\_CFGRX1)

The Transmitter configuration within the SGMII SerDes interface is configured using these registers:

- SGMII Transmitter Configuration Register (SGMII\_SERDES\_CFGTX0 and SGMII\_SERDES\_CFGTX1)

For details on SGMII SerDes configuration registers, definitions and settings, see the *KeyStone Architecture GbE User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

An example SerDes Configuration for SGMII interface is shown below. For the complete suite of configuration examples available from TI, see the Multicore Software Development Kit (MCSDK).

#### Example 4 SGMII SerDes Configuration

```

-----
/*****
* SGMII SERDES CALCULATIONS EXAMPLE
*****/
//NEED linerate of sgmiibyte clock = 1250 MHz
// RATESCALE: FULL - 0.5; HALF - 1; QUATER - 2
//SET sgmiibyte refclkp = 250
//SET MPY = 10 (7'b0101000)
//SET RATE = 2 (2'b01)
// linerate = (refclkp * MPY)/(RATE)
// linerate = (250 * 10)/(2)
// linerate = 2500/2
// linerate = 1250
#define SGMII_SERDES_CFGPLL_VAL 0x00000051 //0x00000041 x8 (multiplier) 0x00000051
x10 (multiplier)
//Bits 0, 3, 5, 9
//31:16 Reserved 0
// 15 STD 0
//14:13 CLKBYP 00
//12:11 LOOP_BWIDTH 0
// 10 SLEEPPLL 0
// 09 VRANGE 0 Vrange set to low as Linerate*RateScale>2170MHz
// 08 ENDIVCLK 0
//07:01 MPY 0101000 --> 10x multiplier, assuming input clock of 250 MHz to
yield the desired 1250 MHz (250 MHz * 10)
// 00 ENPLL 1
#define SGMII_SERDES_CFGRX_VAL 0x00700621
//31:25 Reserved 0000000
//23:24 LOOPBACK 00
// 22 ENOC 1
//21:18 EQ 1100
//17:15 CDR 000 Second order - Phase offset tracking up to +-313 ppm
//14:12 LOS 000
//11:10 ALIGN 01
//09:07 TERM 100
// 06 INVPAIR 0
//05:04 RATE 10 -- tie off (10) //00 = Full Rate , 01 = Half Rate ,
10 = Quarter Rate

```

```

//03:01  BUSWIDTH    000
//    00  ENRX        1
#define SGMII_SERDES_CFGTX_VAL  0x000108A1
//31:22  Reserved    0
//21:20  LOOPBACK    00
//19:18  RDTCT       00
//    17  ENIDL       0
//    16  MYSNC       1
//15:12  DEMPHASIS   0000
//11:08  SWING        1000
//    07  CM          1
//    06  INVPAIR     0
//05:04  RATE         10  -- tie off (10)  //00 = Full Rate , 01 = Half Rate ,
10 = Quarter Rate
//03:01  BUSWIDTH    000
//    00  ENTX        1

/*****
* Setup the SGMII SERDES modules in bootcfg space
*****/
{
*((volatile unsigned int *) 0x02620038) = 0x83E70B13;           //unlock boot
cfg kick0 register
*((volatile unsigned int *) 0x0262003C) = 0x95A4F1E0;           //unlock boot
cfg kick1 register
*((volatile unsigned int *) 0x02620340) = SGMII_SERDES_CFGPLL_VAL; //setup sgmi0
serdes cfgpll
while ((SGMII0_STATUS & 0x00000010) != 0x00000010);           // Wait for the
PLL lock bit to be set
*((volatile unsigned int *) 0x02620344) = SGMII_SERDES_CFGRX_VAL; //setup sgmi0
serdes rx cfg
*((volatile unsigned int *) 0x02620348) = SGMII_SERDES_CFGTX_VAL; //setup sgmi0
serdes tx cfg
*((volatile unsigned int *) 0x0262034C) = SGMII_SERDES_CFGRX_VAL; //setup sgmi1
serdes rx cfg
*((volatile unsigned int *) 0x02620350) = SGMII_SERDES_CFGTX_VAL; //setup sgmi1
serdes tx cfg
*((volatile unsigned int *) 0x02620038) = 0x01234567;           //lock boot cfg
kick0 register
*((volatile unsigned int *) 0x0262003C) = 0x01234567;           //lock boot cfg
kick1 register
}

```

**End of Example 4**

-----

## 7 PCIe Interface

The PCI express (PCIe) module on KeyStone I devices provides an interface between the SoC and other PCIe-compliant devices. This is a two-lane SerDes interface designed to operate up to 5 GBaud per lane from pin-to-pin. The PCI express module provides low pin-count, high-reliability, and high-speed data transfer at rates of 5.0 Gbps per lane on the serial links.

For more details on PCIe interface, see the device-specific data manuals. Specifications supported, interface specific PCB routing recommendations, SerDes device settings accessible by register accesses and are covered in further sections of this document.

### 7.1 Relevant Industry Standard Specification Support

The PCIe interface on KeyStone I devices is compliant with Physical Layer Specifications referenced in Chapter 4 of the PCI Express Base Specification Revision 2.0.

### 7.2 Recommended SerDes PCB Layout Constraints

Routing requirements for the PCIe interface shall adhere to good engineering practices for transmission lines operating above 5 GHz. Specific attention shall be paid to net classes within this group and should have a high routing priority (if this interface is used).

- Each complementary PCIe SerDes receive pair shall be individually skew matched to within 1 ps. 1 ps equates to approximately 5.464 mils to 7.092 mils (depending on propagation delays). Example of complementary pairs include PCIERXN0 & PCIERXP0.
- Both complementary PCIe SerDes receive pairs shall be routed on the same layer.
- Both complementary PCIe SerDes transmit pairs shall be routed on the same layer.
- Both complementary PCIe receive pairs PCIERXN/P1:0 shall be assigned to an individual net class where routing skew shall not be greater than 5 ps between all receive pairs.
- All complementary PCIe transmit pairs PCIETXN/P1:0 shall be assigned to an individual net class and routing skew shall not be greater than 10 ps between all transmit pairs.
- Transmit and receive signals must be referenced to parallel ground planes.
- Vias are allowed and should never exceed 2 per net, all nets must be balanced and the impact of the via on timing, reflections, and loading taken into account during design and layout. This interface should be modeled to assure functionality.
- Differential signal routing must achieve a 100-Ω differential impedance.



**Note**—Vias in PCIe nets are typically not recommended and pose problems if not implemented correctly with regard to signal integrity.

### 7.3 Recommended SerDes Register Configuration Options

Unlike other SerDes interfaces, PCIe configuration registers are spread between Device and MMR registers.

The PLL configuration within the PCIe SerDes interface is configured using these registers:

- The PCIe SerDes Configuration Register (PCIE\_SERDES\_CFGPLL)
- The PCIe SerDes Status Register (PCIE\_SERDES\_STS)

The Receiver configuration within the PCIe SerDes interface is configured using these registers:

- The PCIe SerDes Configuration Lane 0 and Lane 1 registers (SERDES\_CFG0 and SERDES\_CFG1)

The Transmitter configuration within the PCIe SerDes interface is configured using this register:

- The PCIe SerDes Configuration Lane 0 and Lane 1 registers (SERDES\_CFG0 and SERDES\_CFG1)

For details on PCIe SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture PCIe User Guide*, in “[Related Documentation From Texas Instruments](#)” on page 55.

An example SerDes Configuration for PCIe interface is shown below. For the complete suite of configuration examples available from TI, see the Multicore Software Development Kit (MCSDK).

#### Example 5 PCIe SerDes Configuration

```

-----
/*****
* PCIe SERDES CALCULATIONS EXAMPLE
*****/
//NEED linerate of sgmiiclk byte clock = 2500 MHz
// RATESCALE: FULL - 0.5; HALF - 1; QUATER - 2
//SET sgmiiclk_refclkp = 100 (reference clock input = 100MHz)
//SET MPY = 64h (7'b1100100) (multiply factor = 25)
//SET RATE = 1 (2'b01)
//
// linerate = (refclkp * MPY)/(RATE)
// linerate = (100 * 25)/ (1)
// linerate = 2500

#define PCIE_SERDES_CFGPLL_VAL 0x000001C9 //(multiplier 25x)
//31:16 Reserved 0
// 15 STD 0
//14:13 CLKBYP 00
//12:11 LOOP_BWIDTH 0
// 10 SLEEPPLL 0
// 09 VRANGE 0
// 08 ENDIVCLK 1--> enable output of a divide-by-5 of PLL clock
//07:01 MPY 1100100 --> 25x multiplier, assuming input clock of
100 MHz to yield the desired 2500 MHz (100 MHz * 25)
// 00 ENPLL 1--> enable PLL

#define PCIE_SERDES_CFG0_VAL 0x622A0
//31:21 Reserved 000000000000
//20:19 TX_LOOPBACK 00
// 18 TX_MSYNCR 1--> only Lane0 enables the master mode for
synchronization
// 17 TX_CM 1--> enable common mode adjustment
// 16 TX_INVPAIR 0
//15:14 RX_LOOPBACK 00
// 13 RX_ENOC 1 --> enable RX offset compensation
//12:09 RX_EQ 0001--> enable RX adaptive equalization
//08:06 RX_CDR 010--> clock recovery: second order, high precision,

```

```

threshold of 1
//05:03   RX_LOS       100--> enable loss of signal detector, reduced
threshold
//02:01   RX_ALIGN    00
//   00   RX_INVPAIR   0

#define PCIE_SERDES_CFG1_VAL 0x222A0
//31:21   Reserved    0000000000
//20:19   TX_LOOPBACK 00
//   18   TX_MSYNCR   0
//   17   TX_CM       1--> enable common mode adjustment
//   16   TX_INVPAIR   0
//15:14   RX_LOOPBACK 00
//   13   RX_ENOC     1 --> enable RX offset compensation
//12:09   RX_EQ       0001--> enable RX adaptive equalization
//08:06   RX_CDR      010--> clock recovery: second order, high precision,
threshold of 1
//05:03   RX_LOS       100--> enable loss of signal detector, reduced
threshold
//02:01   RX_ALIGN    00
//   00   RX_INVPAIR   0

/*****
*****
* Setup the PCIe SERDES modules in bootcfg space and PCIe MMRs
*****
*****/
{
*((volatile unsigned int *) 0x02620038) = 0x83E70B13;           //unlock
boot cfg kick0 register
*((volatile unsigned int *) 0x0262003C) = 0x95A4F1E0;           //unlock
boot cfg kick1 register

*((volatile unsigned int *) 0x02620358) = PCIE_SERDES_CFGPLL_VAL; //setup PCIe
serdes cfgpll

while ((*((volatile unsigned int *) 0x0262015C) & 0x1) != 0x1); //wait for
the PLL lock bit to be set

*((volatile unsigned int *) 0x21800390) = PCIE_SERDES_CFG0_VAL; //setup
PCIe serdes cfg0 (lane 0)
*((volatile unsigned int *) 0x21800394) = PCIE_SERDES_CFG1_VAL; //setup
PCIe serdes cfg1 (lane 1)

*((volatile unsigned int *) 0x02620038) = 0x01234567;           //lock boot
cfg kick0 register
*((volatile unsigned int *) 0x0262003C) = 0x01234567;           //lock boot
cfg kick1 register
}

```

**End of Example 5**

-----

## 8 SerDes Configuration common to AIF2, SRIO, and HyperLink

### 8.1 SerDes PLL Configuration

The PLL configuration within the AIF SerDes interface is configured using these registers:

- The AIF SerDes PLL Configuration Register (SD\_PLL\_B8\_EN\_CFG, SD\_PLL\_B4\_EN\_CFG, SD\_PLL\_B8\_CFG, SD\_PLL\_B4\_CFG)
- The AIF SerDes PLL Status Register (SD\_PLL\_B8\_STS, SD\_PLL\_B4\_STS)

For details on AIF SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture AIF2 User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

The PLL configuration within the SRIO SerDes interface is configured using this register:

- The SRIO SerDes PLL Configuration Register (SRIO\_SERDES\_CFGPLL)

For details on SRIO SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture SRIO User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

The PLL configuration within the HyperLink SerDes interface is configured using this register:

- The HyperLink SerDes PLL Configuration Register (HYPERLINK\_SERDES\_CFGPLL)

For details on HyperLink SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture HyperLink User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

#### 8.1.1 Enabling the PLL

To enable the internal PLL, the ENPLL bit of PLL configuration register must be set high. After setting this bit, it is necessary to wait for the LOCK bit of PLL Status register to be driven high before RX or TX channels are enabled.

#### 8.1.2 Reference Clock Multiplication

During normal operation the integrated PLL uses  $\text{refclk}/n$  to generate a higher frequency clock from which the bit rate can be derived.  $\text{refclk}/n$  can be in the range 100 - 800 MHz nominal. The clock generated by the PLL will be between 4 and 25 times the frequency of  $\text{refclk}/n$ , according to the multiply factor selected via the MPY field.

#### 8.1.3 VCO Speed Range

It is necessary to adjust the loop filter depending on the operating frequency of the VCO. To indicate the selection the user must set the VRANGE bit. If the VCO is running at the lower end of frequency range the VRANGE should be set high.

If  $\text{LINERATE} \times \text{RATESCALE} < 2.17 \text{ GHz}$ , VRANGE should be set high.

### 8.1.4 Jitter and PLL Loop Bandwidth

Jitter on the reference clock will degrade both the transmit eye and receiver jitter tolerance thereby impairing system performance. A good quality, low jitter reference clock is necessary to achieve compliance with most if not all physical layer standards. To minimize the introduction of additional on-chip jitter the differential reference clock inputs  $\text{refclk}/n$  must be driven from a low jitter clock input buffer (LJCB) or from an off chip LC-based cleaner PLL.

[Table 3](#) summarizes how to select one of the available settings. The setting to use depends on the application and the reference clock frequency. The PLL bandwidth is a function of the reference clock frequency, as shown in the equation in Section 8.1.3 “VCO Speed Range”. The value of BWSCALE is a function of both LB and the PLL output frequency, as shown in [Table 3](#).

**Table 3 PLL Loop Bandwidth Selection (MHz)**

Value	Effect	BWSCALE vs PLL Frequency		
		3.125GHz	2.1GHz	1.5625GHz
00	Medium Bandwidth	13	14	16
01	Ultra High Bandwidth	7	8	8
10	Low Bandwidth	21	23	30
11	High Bandwidth	10	11	14

## 8.2 SerDes Receiver Configuration

The Receiver configuration within the AIF SerDes interface is configured using these registers:

- The AIF SerDes Configuration Register (SD\_RX\_EN\_CFG, SD\_RX\_R1\_CFG, SD\_RX\_R2\_CFG)
- The AIF Receiver SerDes Status Register (SD\_RX\_STS)

For details on AIF SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture AIF2 User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

The Receiver configuration within the SRIO SerDes interface is configured using this register:

- The SRIO SerDes Configuration Register (SRIO\_SERDES\_CFGRXn)

For details on SRIO SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture SRIO User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

The Receiver configuration within the HyperLink SerDes interface is configured using these registers:

- The HyperLink SerDes Configuration Register (HYPERLINK\_SERDES\_CFGRXn)
- The HyperLink Receiver SerDes Status Register (HYPERLINK\_SERDES\_STS)

For details on HyperLink SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture HyperLink User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

### 8.2.1 Data Rate Selection

The primary operating frequency of the SerDes macro is determined by the reference clock frequency and PLL multiplication factor. However, to support lower frequency applications, each receiver can also be configured to operate at a half, quarter or eighth of this rate via the RATE bits as described in [Table 4](#).

**Table 4 Receiver Operating Rate**

Value	Rate
00	Full rate. Four data samples taken per PLL output clock cycle.
01	Half rate. Two data sample taken per PLL output clock cycle.
10	Quarter rate. One data sample taken per PLL output clock cycle.
11	Eighth rate. One data sample taken every two PLL output clock cycles.



**Note**—The AIF2 RX interface does not support full rate mode.

### 8.2.2 Receiver Termination

The rxpi and rxni differential inputs are each internally terminated to a common point via a 50-Ω resistor, i.e., a 100-Ω differential termination. This TERM value should be set to 001: Common point set to 0.7 VDDT for all use cases. Common point set to 0.7 VDDT is the recommended configuration for all AC coupled systems.

Note that when AC coupled, the data carried by the transmission media should be DC balanced to ensure baseline wander is avoided (for example by using a coding scheme such as 8b:10b). Failure to avoid baseline wander will result in increased jitter in the data stream at the receiver and may cause data to be lost.

### 8.2.3 Adaptive Equalizer

The receiver incorporates an adaptive equalizer, which can compensate for channel insertion loss by attenuating the low frequency components with respect to the high frequency components of the signal, thereby reducing inter-symbol interference. When enabled, the receiver equalization logic analyzes data patterns and transition times to determine whether the low frequency gain should be increased or decreased.

The decision logic is implemented as a voting algorithm with a relatively long analysis interval. The slow time constant that results reduces the probability of incorrect decisions but allows the equalizer to compensate for the relatively stable response of the channel. [Table 5](#) and [Table 6](#) includes details on Equalizer Configuration and Hold settings.

**Table 5 Receiver Equalizer Configuration (Part 1 of 2)**

EQ[2:0]	Effect
000	No equalization. The equalizer provides a flat response at the maximum gain. This setting may be appropriate if jitter at the receiver occurs predominantly as a result of crosstalk rather than frequency dependent loss.
001	Fully adaptive equalization. The zero position is determined by the selected operating rate, and the low frequency gain of the equalizer is determined algorithmically by analyzing the data patterns and transition positions in the received data. This setting should be used for most applications.

**Table 5 Receiver Equalizer Configuration (Part 2 of 2)**

EQ[2:0]	Effect
010	Precursor equalization analysis. The data patterns and transition positions in the received data are analyzed to determine whether the transmit link partner is applying more or less precursor equalization than necessary.
011	Postcursor equalization analysis. The data patterns and transition positions in the received data are analyzed to determine whether the transmit link partner is applying more or less postcursor equalization than necessary.
<b>End of Table 5</b>	

**Table 6 Receiver Equalizer Hold**

EQHLD	Effect
0	Equalizer adaptation enabled. The equalizer adaptation and analysis algorithm is enabled. This should be the default state.
1	Equalizer adaptation held. The equalizer is held in it's current state. In addition, the adaptation and analysis algorithm is reset.

The adaptive algorithm works by analyzing data and transition samples of single bits after 3 or 4 bits without transition. It breaks the received data into 5 bit groups, at all possible alignments (i.e. each group contains 4 bits also present in the previous group). Within each group, it looks at the transition between the 4th and 5th bits for the patterns of interest to determine whether the data is over or under equalized.

The algorithm relies on the fact that over a short period of time, the clocks cannot be both early and late. Hence the combinations early, under equalized and late, under equalized indicate that the data is under equalized. Similarly the combinations early, over equalized and late over equalized indicate that the data is over equalized.

Whenever there is an unambiguous indication of under or over equalization within a 16 bit group, a 256 bit counter is incremented or decremented respectively. Whenever the counter overflows or underflows, the equalizer setting is adjusted.

### 8.2.3.1 Pre and Post Cursor Equalization Analysis

When EQ is set to 010 or 011, the equalizer is reconfigured to provide analytical data about the amount of pre and post cursor equalization respectively present in the received signal. This can in turn be used to adjust the equalization settings of the transmitting link partner, where a suitable mechanism for communicating this data back to the transmitter exists.

Status information is provided via RX Status register bit-fields (EQOVER, EQUNDER), by using the following method:

1. Enable the equalizer by setting EQHLD low and EQ to 001. Allow sufficient time for the equalizer to adapt.
2. Set EQHLD to 1 to lock the equalizer and reset the adaptation algorithm. This also causes both EQOVER and EQUNDER to become low.
3. Wait at least 48UI, and proportionately longer if the CDR activity is less than 100%, to ensure the 1 on EQHLD is sampled and acted upon.
4. Set EQ to 010 or 011, and EQHLD to 0. The equalisation characteristics of the received signal are analyzed (the equalizer response will continue to be locked).
5. Wait at least 150103UI to allow time for the analysis to occur, proportionately longer if the CDR activity is less than 100%.
6. Examine EQOVER and EQUNDER for results of analysis.

- a. If EQOVER is high, it indicates the signal is over equalized.
- b. If EQUNDER is high, it indicates the signal is under equalized.
7. Set EQHLD to 1.
8. Repeat items 3 - 7 if required.
9. Set EQ to 001, and EQHLD to 0 to exit analysis mode and return to normal adaptive equalization.

Note that when changing EQ from one non-zero value to another, EQHLD must already be 1. If this is not the case, there is a chance the equalizer could be reset by a transitory input state (i.e. if EQ is momentarily 000). EQHLD can be set to 0 at the same time as EQ is changed.

As the equalizer adaptation algorithm is designed to equalize the post cursor, EQOVER or EQUNDER will be set only during post cursor analysis if the amount of post cursor equalization required is more or less than the adaptive equalizer can provide.

However, this should be independently configurable on a per channel basis. Offset compensation is required to obtain best receiver performance. This is enabled by setting ENOC high, which should be the default setting. The ability to turn this off should be provided, and can be shared across all similarly configured receivers.



**Note**—For AIF (OBSAI and CPRI), No equalization mode (setting 000) is appropriate and EQHOLD should be set to zero. HyperLink has adaptive equalization enabled at all times.

### 8.2.4 Differential Sense Amplifiers

Data input on rxpi and rxni (where i= lane1, lane2, lane3 and so on) is sampled by differential sense amplifiers using clocks derived by the clock recovery algorithm. Each bit must be sampled twice, half a bit period apart, to provide the necessary information for the clock recovery algorithm. Furthermore, the period of the PLL output clock is equal to four bit periods during full rate operation, requiring eight identical samplers, plus a further two for eyescan and eye height analysis.

The polarity of rxpi and rxni can be inverted by setting the INVPAIR bit as show in [Table 7](#). This can potentially simplify PCB layout and improve signal integrity by avoiding the need to swap over the differential signal traces so that txpi connects to rxpi and txni connects to rxni.

**Table 7 Receiver Polarity**

INVPAIR	Effect
0	Normal polarity. rxpi considered to be positive data and rxni negative.
1	Inverted polarity. rxpi considered to be negative data and rxni positive.

Due to processing effects, the devices in the rxpi and rxni differential sense amplifiers will not be perfectly matched and there will be some offset in switching threshold. The macro contains circuitry to detect and correct for this offset. This feature can be enabled by setting the ENOC bit as show in [Table 8](#).

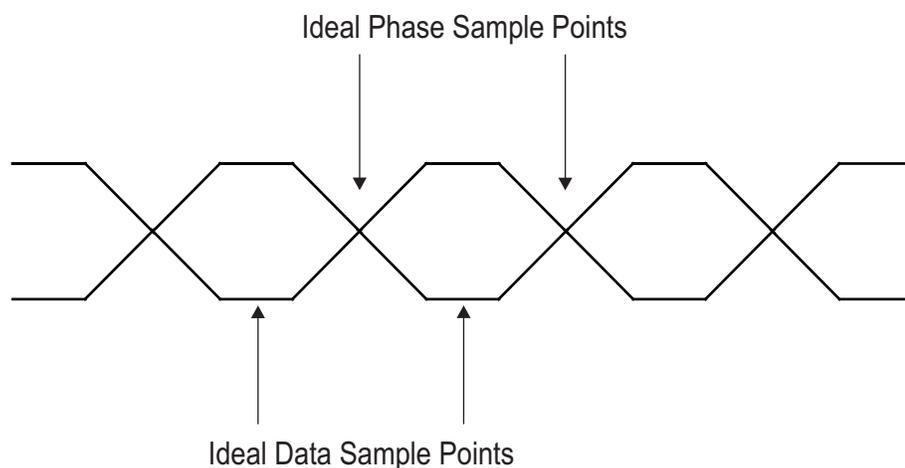
**Table 8 Offset Compensation**

ENOC	Effect
0	Compensation disabled
1	Compensation enabled.

### 8.2.5 Clock Recovery Algorithms

The clock recovery algorithms operate to adjust the clocks used to sample rxpi and rxni so that the data samples are taken midway between data transitions, as shown in [Figure 1](#). Both first and second order algorithms are provided. The second order algorithm can be optionally disabled, and both can be configured to optimize their dynamics.

**Figure 1 Receiver Data and Phase Samples**



Both algorithms use the same basic technique for determining whether the sampling clock is ideally placed, and if not whether it needs to be moved earlier or later. When two contiguous data samples are different, the phase sample between the two is examined. The sampling clock can be considered early or late depending on whether the phase sample matches the first or second data sample respectively. Twelve such comparisons are made from contiguous bits (i.e. twelve data samples and thirteen phase samples) with each result counted as a vote to move the sample point either earlier or later. These twelve data bits constitute the voting window. The twelve votes are then summed, and an action to adjust the position of the sampling clock occurs if the difference between early and late votes exceeds a programmable threshold. In addition, the first order algorithm can be configured to operate at a reduced activity level, to save power at the expense of phase tracking rate. [Table 9](#) summarizes how to select the available options via CDR. These options are discussed in more detail in the following two sections.

**Table 9 Clock Data Recovery Algorithm selection**

Value	Vote Threshold	Tracking Rate (ppm)	Order	Settling Time (UI)	Effective Activity (%)
000	15	313	2nd	36	83
001	7	607			70
010	3	723			50
011	1	868			25
100	15	96	1st	1524	83
101	3	289			50
110	1	434			25
111	7	13			5

The first order algorithm makes a single phase adjustment each time the 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.

Setting 110 being the default in synchronous applications (that is, where the same clock source is used at both end of the link). If the crystals are different, there will be a constant ppm offset between the two ends of the link, and in this case it is best to enable the second order loop (that is, setting 011 as default). Options that use increased voting thresholds may offer improved jitter tolerance performance in some applications, but this is difficult to predict (really need to perform a detailed Matlab-type analysis, or try it empirically). Setting 111 should be used only in short-reach synchronous chip-to-chip applications where squeezing out as much power as possible.

### 8.2.5.1 First Order Clock Recovery Algorithm

The majority vote from each voting window is used to adjust a signed counter (increment if majority late, decrement if majority early, no change if a tie). Whenever the magnitude of the counter reaches the specified threshold, the sampling position is adjusted by 1 / 48 UI. The results from the subsequent three voting windows are then ignored to allow the control loop to settle. Additional voting windows are ignored when CDR = 111, to save power.

If the threshold is set to 1, this results in a maximum tracking rate of 1 / 48 UI per 48 bits,  $\pm 434$  ppm (1 / (48 × 48)). If the threshold is set to 15, this results in a maximum tracking rate of  $\pm 96$  ppm (1 / (48 × 216)). The actual tracking rate will be lower than this if the average transition density of the data is less than 1 in 12. In general, the tracking rate can be expressed mathematically as shown in equation 1, where S is the settling time, T is the threshold and D is the transition density. The multiplicand in the denominator is rounded up to the nearest 12 UI, as this is the size of the voting window.

**Equation 1. 1<sup>st</sup> Order CDR Tracking Rate**

$$\text{Tracking Rate} = \frac{1}{48 \times ((R)ND_{12}(1/D) \times T) + S)} \text{ppm}$$

This can be used to consider both typical and worst case scenarios. For example:

- If T is 15, S is 1524 UI, and the average transition density is 1 in 5, it will take  $RND_{12}(5) \times 7 = 12 \times 7 = 84$  UI to reach the threshold. So, average tracking rate would be 1 / (48 × (84 + 1524)) = 13 ppm.

- If T is 1, S is 36 UI, and the worst case transition density is 1 in 70, it will take 70 UI to reach the threshold, which rounds up to 72 UI. So, worst case tracking rate would be  $1 / (48 \times (72 + 36)) = 192$  ppm.

Even so, this analysis is slightly optimistic, as it assumes there are no incorrect votes. These may occur in the presence of jitter, and may affect how long it takes to exceed the selected threshold. For a worst case analysis, any such optimism is generally offset by the fact that the worst case pattern (edges at minimum transition density occurring indefinitely with no higher density periods) is a pathological case that does not actually occur.

It is important to ensure that the time between votes being taken and the phase adjustment occurring is kept to a minimum. If possible, it should be kept within the 16-UI blanking interval in order to minimize voting jitter in the first order algorithm.

Worst case lock time requires the phase of the sampling clocks to be adjusted by 1 2UI. How long this takes will depend on the amount of underlying frequency offset that needs to be tracked. This is expressed mathematically in equation 2 (where tracking rate is defined by “[Equation 1. 1st Order CDR Tracking Rate](#)” on page 30).

**Equation 2. Lock time (UI)**

$$T_{\text{LOCK}} (\text{rxpirxni}) = \frac{5 \times 10^3}{\text{Tracking Rate} - \text{frequency offset (ppm)}} \text{ UI}$$

### 8.2.5.2 Second Order Clock Recovery Algorithm

The second order algorithm keeps a running total of the net number of times the selected first order threshold is equalled or exceeded in a signed accumulator. An excess of late votes increment the accumulator, and excess of early votes decrement it. If the accumulator is positive, regular positive phase adjustments will be made, and if negative, regular negative phase adjustments will be made. The larger the magnitude of the accumulator, the more frequent the phase adjustments will be. When the number of regular adjustments being made is enough to compensate for the underlying frequency offset, the number of early and late vote excesses will balance on average, and the accumulator value will stabilize at a value proportional to the frequency offset. Short-term phase changes will be tracked by the first order algorithm, which continues to operate as normal.

The second order algorithm can compensate for phase offsets up to 434 ppm, with a precision of 1.7 ppm (434 / 255). The maximum tracking rate is independent of the first order setting, although the lock time of the second order loop is inversely proportional to the first order tracking rate.

It is recommended that the second order algorithm be used in all asynchronous applications, even when the required tracking rate is well within the capabilities of the first order algorithm. By preemptively compensating for any underlying frequency offset using the second order loop, the sampling point will be closer to the centre of the data eye and bit error rate performance will be improved.

### 8.2.6 Jitter Tolerance

The incoming data stream may contain several forms of jitter, which must be filtered by the receiver to ensure successful reception of the data. High frequency data-dependent jitter caused by channel insertion loss will be compensated for by the adaptive equalizer, which precedes all other receive circuitry in the data path.

High frequency jitter that exhibits a probability distribution with zero mean will be rejected by the first order clock recovery algorithm, which applies an averaging technique to detected edge positions to determine the ideal data sample point.

Low frequency jitter, which may be sinusoidal or due to constant frequency error, will also be filtered by the clock recovery algorithm tracking function and limited only by the maximum tracking rate. These different components of jitter in the receive eye should be assessed individually to determine the overall jitter tolerance.

### 8.2.7 Serial to Parallel conversion

Data sampled on rxpi and rxni is de serialized into 16- or 20-bit symbols according to the BUSWIDTH field of cfgrxi, as shown in [Table 10](#).

**Table 10 Bus Width**

BUSWIDTH	Effect
010	20-bit operation
011	16-bit operation



**Note**—The receiver bus width is fixed in hardware to 20 bits and is not configurable through an MMR for AIF, HyperLink and SRIO.

### 8.2.8 Symbol Alignment

Each receiver independently supports symbol alignment, selectable via the ALIGN field of cfgrxi register as shown in [Table 11](#).

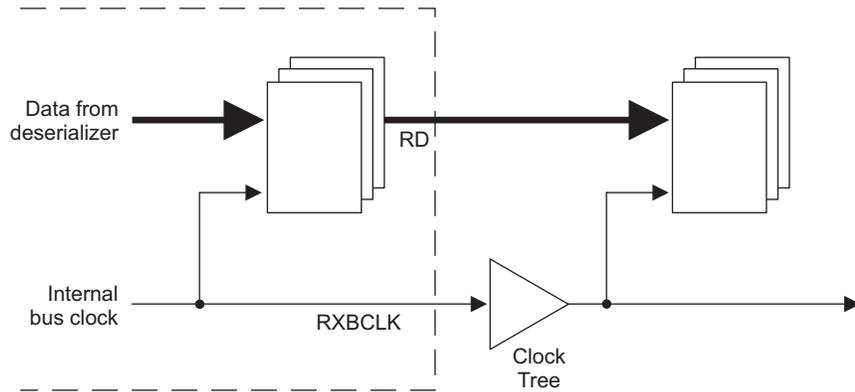
**Table 11 Receiver Symbol Alignment Selection**

ALIGN	Effect
00	Alignment Disabled. No symbol alignment will be performed whilst this setting is selected, or when switching to this selection from another.
01	Comma alignment enabled. Symbol alignment will be performed whenever a misaligned comma symbol is received.
10	Alignment Jog. The symbol alignment will be adjusted by one bit position when this mode is selected.
11	Reserved

### 8.2.8.1 Comma-Based Alignment

Setting the ALIGN field to 01 enables alignment to the K28 comma symbols included in the 8b:10b data encoding scheme defined by the IEEE and employed by numerous transmission standards.

**Figure 2** rcbclk[i] and rdi[19:0] Interface



In this mode, the receiver examines the data stream for the positive or negative disparity comma characters (0011111xxx and 1100000xxx respectively). If a comma that straddles symbol boundaries is detected, the alignment will be corrected by adjusting the relationship between rxbclk[i] and rdi[19:0] appropriately. The realignment process can take up to 51 rxbclk[i] periods to complete, during which the data output on rdi[19:0] will be corrupted, and the period of rxbclk[i] will be slightly increased.

Furthermore, if the 3 bits preceding a misaligned comma are the same as the MSB of the comma, it will be ignored. This prevents incorrect re-alignment happening in applications that use K28.7 characters. (A K28.7 followed by certain other symbols can produce what looks like a comma pattern across the symbol boundary. The K28.7 will be aligned to, but not the spurious comma that follows.)

Whilst comma detection is enabled, the SYNC bit of RX Status register stsrxi will be driven high synchronously to rxbclk[i] the cycle after a comma is output on rdi[19:0], provided that no realignment was required (i.e. the comma was already correctly aligned).

Comma re-alignment is not intended for use in systems that do not contain comma symbols (for example, unencoded 8-bit data symbols). As such, comma alignment is performed only if enabled when BUSWIDTH = 010.

Note that the entire family of K28 symbols are acted upon (i.e. the 3 LSBs are ignored).

When comma detection is disabled, the present alignment will be maintained, and SYNC will remain low. However, as a result of internal pipelining, it is possible for a comma received just before disabling to cause either a realignment or SYNC to activate.

### 8.2.8.2 Single Bit Alignment Jog

For systems that cannot use comma-based symbol alignment, the single bit alignment jog capability provides a means to control the symbol realignment features of the receiver directly from logic implemented in the ASIC core. This logic can be designed to support whatever alignment detection protocol is required.

When a rising edge is detected on the MSB of ALIGN, a single bit adjustment to the alignment will be made (one data bit will be lost), in exactly the same way as for a comma misaligned by one bit.

Realignment will occur within 15 rxblk[i] cycles of a rising edge being sampled on the MSB of ALIGN.

SYNC will be driven high for the first rxblk[i] cycle during which realigned data is output on rdi[9:0]. The latency between applying a rising edge to the MSB of ALIGN and the alignment completing means there is a potential for *realignment overshoot*. The easiest way to avoid this is to wait for SYNC to be driven high before asserting the MSB of ALIGN again.

### 8.2.9 Loss of Signal Detection

Each receive channel also supports loss of signal detection, configured via the LOS bits of cgrxi register. The LOSi bits should be set to 2b10 to allow Loss of signal detection by the SerDes macro. For internal loopback, this should be disabled.

## 8.3 SerDes Transmitter Configuration

The Transmitter configuration within the AIF SerDes interface is configured using these registers:

- The AIF SerDes Configuration Register (SD\_TX\_EN\_CFG, SD\_TX\_R1\_CFG, SD\_TX\_R2\_CFG)
- The AIF Transmitter SerDes Status Register (SD\_TX\_STS)

For details on AIF SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture AIF2 User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

The Transmitter configuration within the SRIO SerDes interface is configured using this register:

- The SRIO SerDes Configuration Register (SRIO\_SERDES\_CFGTXn)

For details on SRIO SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture SRIO User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

The Transmitter configuration within the HyperLink SerDes interface is configured using these registers:

- The HyperLink SerDes Configuration Register (HYPERLINK\_SERDES\_CFGTXn)
- The HyperLink Transmitter SerDes Status Register (HYPERLINK\_SERDES\_STS)

For details on HyperLink SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture HyperLink User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55.

### 8.3.1 Data Rate Selection

The primary operating frequency of the SerDes macro is determined by the reference clock frequency and PLL multiplication factor. For example, AIF2 uses the system clock of 307.2 MHz generated by one of the transmit links. All the links must be set to 8× to activate any internal module link rate (2×, 4×, 5×, 8×) and at least this field should be set to 0x1 to activate other module before activating TX SerDes. This field should be set to Half rate(8×) for all six links at initialization time before locking the SerDes PLL.



**Note**—The AIF2 TX interface supports only the half rate mode.

### 8.3.2 MSYNC

MSYNC field setting is only for the AIF Interface. See the *KeyStone Architecture AIF2 User Guide*, in [“Related Documentation From Texas Instruments”](#) on page 55 for details on MSYNC setting.

### 8.3.3 Serial to Parallel Conversion

The receiver bus bandwidth is fixed in hardware to 20 bits and is not configurable through an MMR.

### 8.3.4 Differential Voltage-Mode Driver

The differential voltage-mode driver takes each bit from the parallel to serial converter and drives txpi and txni outputs accordingly. The polarity of txpi and txni can be inverted by setting the INVPAIR bit of cfgtxi registers.

### 8.3.5 Output Voltage Swing Control

The output swing of the differential driver of each transmitter can be independently set to one of a number of settings via the SWING bits of cfgtxi register. Reducing the output amplitude decreases the current drawn from VDDR in direct proportion to the reduction in swing, thereby saving power.

The standard specifies several compliant interconnect and associated far-end eye templates. The swing and equalization settings required depends on which of these templates is being targeted by the application. If the equalization function is not used, these setting should be 0000 for all.

### 8.3.6 Pre and Post Cursor FIR filter

The transmitted waveform has compensation for high frequency attenuation in the attached media via an FIR filter with up to one pre-cursor and one post-cursor taps. Configuration is set up as follows:

- TWPRE taps can contribute 0 to -17.5% in 2.5%steps
- TWPST taps can contribute -37.5 to +37.5% in 2.5% steps

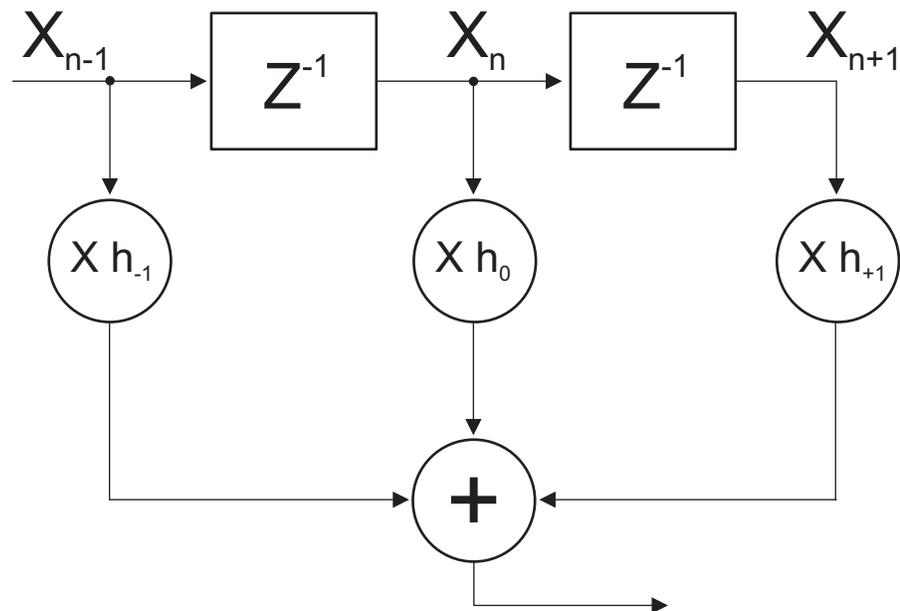
Many combinations of tap weights will lead to undesired filter configurations. The correct filter configuration must be evaluated for the application link. In applications where the FIR codes need to be changed while transmitting data, changes should be made when FIRUPT is low. The rising edge of FIRUPT can then clock the changes into the transmitter. In applications where the FIR codes do not need to change without disturbing the transmitted data, FIRUPT must be tied high. This is generally the case for data rates below 6 Gbps.

Because FIRUPT (ANDed with txblk) controls a transparent latch, in applications where it is not tied high, FIR codes should not change in the cycle FIRUPT is high, or else they must be re-timed to the falling edge of txblk to avoid hold issues.

Note that while the PLL is out of lock, the transmit FIR coefficients are continuously updated irrespective of the state of cfgtxi.

Figure 3 shows the structure of the 3 tap FIR comprising, pre-cursor, main cursor and post cursor taps. The weights  $h_{-1}$  and  $h_{+1}$  are set according to Table 12 and Table 13 respectively.

**Figure 3** Transmitter FIR Block Diagram



**Table 12** Precursor Transmit Tap Weights

TWPRE	
Value	Tap weight (%)
000	0
001	-2.5
010	-5.0
011	-7.5
100	-10.0
101	-12.5
110	-15.0
111	-17.5

**Table 13 Transmit Postcursor 1 Tap Weights (Nominal)**

TWPST1			
Value	Pos tap weight (%)	Value	Neg tap weight (%)
00000	0	10000	0
00001	+2.5	10001	-2.5
00010	+5.0	10010	-5.0
00011	+7.5	10011	-7.5
00100	+10.0	10100	-10.0
00101	+12.5	10101	-12.5
00110	+15.0	10110	-15.0
00111	+17.5	10111	-17.5
01000	+20	11000	-20
01001	+22.5	11001	-22.5
01010	+25.0	11010	-25.0
01011	+27.5	11011	-27.5
01100	+30	11100	-30
01101	+32.5	11101	-32.5
01110	+35.0	11110	-35.0
01111	+37.5	11111	-37.5
<b>End of Table 13</b>			

In order to calculate post-cursor de-emphasis as a percentage of the main cursor, it is necessary to understand that the implementation is constrained such that  $h_0 = 1 - |h_{+1}| - |h_{-1}|$ . For example, assuming no pre-cursor correction is applied, a 25% post cursor tap weight results in a main cursor weight of 75%. The de-emphasis following a transition is therefore given by:

$$\frac{75\% - 25\%}{75\% + 25\%} = 50\% = -6 \text{ dB}$$

Note that if the sum of the pre-cursor and post-cursor tap weights reaches 50%, the numerator of this expression reduces to 0, and the signal collapses. Care should be taken to ensure the cursor weight does not fall below 50%.

## 9 SerDes Configuration Common to PCIe and SGMII

### 9.1 SerDes PLL Configuration

The PLL configuration within the PCIe SerDes interface is configured using these registers:

- PCIe SerDes PLL Configuration Register (PCIE\_SERDES\_CFGPLL)
- The PCIe SerDes Status Register (PCIE\_SERDES\_STS).

For details on PCIe SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture PCIe User Guide*, in “[Related Documentation From Texas Instruments](#)” on page 55.

The PLL configuration within the SGMII SerDes interface is configured using this register:

- SGMII SerDes PLL Configuration Register (SGMII\_SERDES\_CFGPLL).

For details on PCIe SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture GbE User Guide*, in “[Related Documentation From Texas Instruments](#)” on page 55.

#### 9.1.1 Enabling The PLL

To enable the internal PLL, the ENPLL bit of PLL configuration register must be set high. After setting this bit, it is necessary to wait for the LOCK bit of PLL Status Register to be driven high before RX or TX channels are enabled.

#### 9.1.2 Reference Clock Multiplication

During normal operation, the integrated PLL uses refclkp/n to generate a higher frequency clock from which the bit rate can be derived. refclkp/n can be in the range of 100 MHz to 800 MHz nominal. The clock generated by the PLL will be between 4 and 25 times the frequency of refclkp/n, according to the multiply factor selected via the MPY field.

#### 9.1.3 VCO Speed Range

It is necessary to adjust the loop filter depending on the operating frequency of the VCO. To indicate the selection, the user must set the VRANGE bit. If the VCO is running at the lower end of frequency range the VRANGE should be set high.

If  $\text{LINERATE} \times \text{RATESCALE} < 2.17 \text{ GHz}$ , VRANGE should be set high.

#### 9.1.4 Jitter and PLL Loop Bandwidth

Jitter on the reference clock will degrade both the transmit eye and receiver jitter tolerance, thereby impairing system performance. A good quality, low jitter reference clock is necessary to achieve compliance with most, if not all, physical layer standards. To minimize the introduction of additional on-chip jitter, the differential reference clock inputs refclkp/n must be driven from a low jitter clock input buffer (LJCB) or from an off-chip LC-based cleaner PLL.

Table 14 summarizes how to select one of the available settings. The setting to use depends on the application and the reference clock frequency. The PLL bandwidth is a function of the reference clock frequency, as shown in the equation in Section 9.1.3 “VCO Speed Range”. The value of BWSCALE is a function of both LB and the PLL output frequency, as shown in Table 14.

**Table 14 PLL Loop Bandwidth Selection (MHz)**

Value	Effect	BWSCALE vs PLL Frequency		
		3.125 GHz	2.1 GHz	1.5625 GHz
00	Medium Bandwidth	13	14	16
01	Ultra High Bandwidth	7	8	8
10	Low Bandwidth	21	23	30
11	High Bandwidth	10	11	14
<b>End of Table 14</b>				

## 9.2 SerDes Receiver Configuration

The Receiver configuration within the PCIe SerDes interface is configured using this register:

- The PCIe SerDes Configuration Lane 0 and Lane 1 registers (SERDES\_CFG0 and SERDES\_CFG1)

For details on PCIe SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture PCIe User Guide*, in “[Related Documentation From Texas Instruments](#)” on page 55.

The Receiver configuration within the SGMII SerDes interface is configured using these registers:

- SGMII Receiver Configuration Register 0 (SGMII\_SERDES\_CFGRX0)
- SGMII Receiver Configuration Register 1 (SGMII\_SERDES\_CFGRX1)

For details on SGMII SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture GbE User Guide*, in “[Related Documentation From Texas Instruments](#)” on page 55.



**Note**—The PCIe interface uses fixed default values for the SERDES\_CFG0 and SERDES\_CFG1 registers.

### 9.2.1 Data Rate Selection

The primary operating frequency of the SerDes macro is determined by the reference clock frequency and PLL multiplication factor. However, to support lower frequency applications, each receiver can also be configured to operate at a half, quarter, or one-eighth of this rate via the RATE bits as described in [Table 15](#).

**Table 15 Receiver Operating Rate**

Value	Rate
00	Full rate. Four data samples taken per PLL output clock cycle.
01	Half rate. Two data sample taken per PLL output clock cycle.
10	Quarter rate. One data sample taken per PLL output clock cycle.
11	Eighth rate. One data sample taken every two PLL output clock cycles.



**Note**—The SGMII interface supports only the half rate mode.

### 9.2.2 Receiver Termination

The rxpi and rxni differential inputs are each internally terminated to a common point via a 50-Ω resistor i.e. 100-Ω differential termination.

Note that when AC coupled, the data carried by the transmission media should be DC balanced to ensure baseline wander is avoided (for example by using a coding scheme such as 8b:10b). Failure to avoid baseline wander will result in increased jitter in the data stream at the receiver and may cause data to be lost.



**Note**—For SGMII the TERM is fixed at 100b i.e. Common point set to VSST. For PCIE leave TERM at its default value.

### 9.2.3 Adaptive Equalizer

The receiver incorporates an adaptive equalizer, which can compensate for channel insertion loss by attenuating the low frequency components with respect to the high frequency components of the signal, thereby reducing inter-symbol interference. When enabled, the receiver equalization logic analyzes data patterns and transition times to determine whether the low frequency gain should be increased or decreased.

The decision logic is implemented as a voting algorithm with a relatively long analysis interval. The slow time constant that results reduces the probability of incorrect decisions but allows the equalizer to compensate for the relatively stable response of the channel.

- No equalization (000) setting provides a flat response at the maximum gain. This setting may be appropriate if jitter at the receiver occurs predominantly as a result of crosstalk rather than frequency dependent loss.
- Fully adaptive equalization setting (001) could be appropriate for most applications. The zero position is determined by the selected operating rate, and the low frequency gain of the equalizer is determined algorithmically by analyzing the data patterns and transition positions in the received data.

- Partially adaptive equalization. The low frequency gain of the equalizer is determined algorithmically by analyzing the data patterns and transition positions in the received data. The zero position is fixed in one of eight zero positions. For any given application, the optimal setting is a function of the loss characteristics of the channel and the spectral density of the signal as well as the data rate, which means it is not possible to identify the best setting by data rate alone, although generally speaking, the lower the line rate, the lower the zero frequency that will be required.

When enabled, see [Table 16](#) the receiver equalization logic analyzes data patterns and transition times to determine whether the low frequency gain should be increased or decreased. For the fully adaptive setting (EQ = 0001), if the low frequency gain reaches the minimum value, the zero frequency is then reduced. Likewise, if it reaches the maximum value, the zero frequency is then increased.

**Table 16 Receiver Equalizer Configuration (EQ)**

EQ Bits	Low-Frequency Gain	Zero-Frequency (at $e_{28}$ (min))
0000	Maximum	-
0001	Adaptive	
0010	Reserved	
0011	Reserved	
0100	Reserved	
0101	Reserved	
0110	Reserved	
0111	Reserved	
1000	Adaptive	365 MHz
1001	Adaptive	275 MHz
1010	Adaptive	195 MHz
1011	Adaptive	140 MHz
1100	Adaptive	105 MHz
1101	Adaptive	75 MHz
1110	Adaptive	55 MHz
1111	Adaptive	50 MHz
<b>End of Table 16</b>		



**Note**—Leave the PCIe register at default settings.

It is prudent to ensure that the low frequency gain is not set to one of the extreme settings (and in particular, the maximum setting) when adaptation starts, as this carries the risk that the initial equalization is too far from the ideal setting to allow the CDR to lock, which is a prerequisite for the equalizer adaptation algorithm. To achieve this, the low frequency gain should be forced to the centre setting when EQ changes from 0000 to one of the other codes, or if the PLL is not locked and EQ is non zero.

The decision logic is implemented as a voting algorithm with a relatively long analysis interval. The slow time constant that results reduces the probability of incorrect decisions but allows the equalizer to compensate for the relatively stable response of the channel.

### 9.2.4 Differential Sense Amplifiers

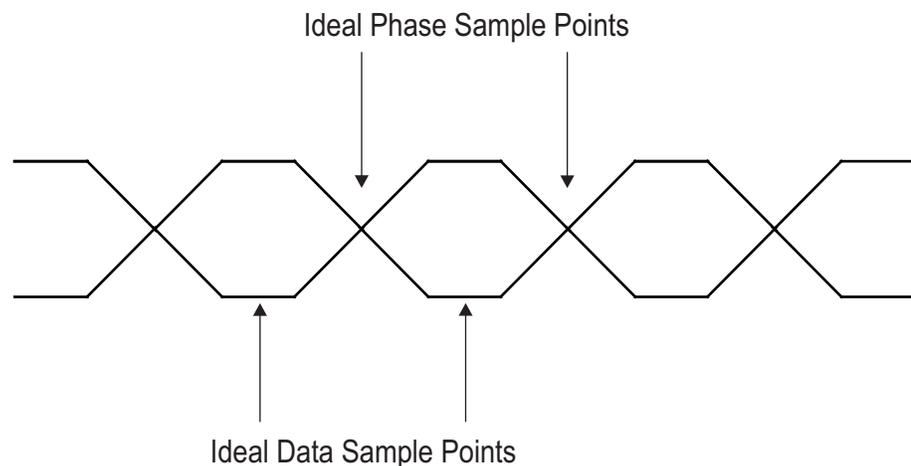
Data input on rxpi and rxni (where  $i = \text{lane1, lane2, lane3}$  and so on) is sampled by differential sense amplifiers using clocks derived by the clock recovery algorithm. Each bit must be sampled twice, half a bit period apart, to provide the necessary information for the clock recovery algorithm. Furthermore, the period of the PLL output clock is equal to four bit periods during full-rate operation, requiring eight identical samplers, plus a further two for eyescan and eye height analysis.

The polarity of rxpi and rxni can be inverted by setting the INVPAIR bit. This can potentially simplify PCB layout and improve signal integrity by avoiding the need to swap over the differential signal traces so that txpi connects to rxpi and txni connects to rxni.

### 9.2.5 Clock Recovery Algorithms

The clock recovery algorithms operate to adjust the clocks used to sample rxpi and rxni so that the data samples are taken midway between data transitions, as shown in [Figure 4](#). Both first and second order algorithms are provided. The second order algorithm can be optionally disabled, and both can be configured to optimize their dynamics.

**Figure 4 Receiver Data and Phase Samples**



Both algorithms use the same basic technique for determining whether the sampling clock is ideally placed, and if not whether it needs to be moved earlier or later. When two contiguous data samples are different, the phase sample between the two is examined. The sampling clock can be considered early or late depending on whether the phase sample matches the first or second data sample respectively. Eight such comparisons are made from contiguous bits (i.e. eight data samples and nine phase samples) with each result counted as a vote to move the sample point either earlier or later. These eight data bits constitute the voting window. The eight votes are then summed, and an action to adjust the position of the sampling clock occurs if the difference between early and late votes exceeds a programmable threshold. The nature of this action will depend on the algorithm selected.

Table 17 summarizes how to select the available options via CDR. These options are discussed in more detail in the following two sections.

**Table 17 Clock Data Recovery Algorithm Selection**

CDR	
Value	Effect
0 0 0	<b>First order</b> , threshold of 1. Phase offset tracking up to $\pm 488$ ppm. Suitable for use in asynchronous systems with low frequency offset.
0 0 1	<b>First order</b> , threshold of 17. Phase offset tracking up to $\pm 325$ ppm. Suitable for use in synchronous systems. Offers superior rejection of random jitter, but is less responsive to systematic variation such as sinusoidal jitter.
0 1 0	<b>Second order</b> , high precision, threshold of 1. Highest precision frequency offset matching but relatively poor response to changes in frequency offset, and long lock time. Suitable for use in systems with fixed frequency offset.
0 1 1	<b>Second order</b> , high precision, threshold of 17. Highest precision frequency offset matching but poorest response to changes in frequency offset, and longest lock time. Suitable for use in systems with fixed frequency offset and low systematic variation.
1 0 0	<b>Second order</b> , low precision, threshold of 1. Best response to changes in frequency offset and fastest lock time, but lowest precision frequency offset matching. Suitable for use in systems with spread spectrum clocking.
1 0 1	<b>Second order</b> , low precision, threshold of 17. Good response to changes in frequency offset and fast lock time, but low precision frequency offset matching. Suitable for use in systems with spread spectrum clocking.
1 1 x	Reserved
<b>End of Table 17</b>	

If contiguous data samples are the same, the transition sample between them should be checked. If it is different, this indicates that in fact there were two transitions. If this occurs, and there is no majority vote, a vote should be forced. This is to counter a pathological case in which a clock pattern with duty cycle error is applied, and the data samples are taken almost on the transitions.

The first order algorithm makes a single phase adjustment each time the 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.

Note that if the loss of signal detector is enabled and there is no signal present at rxpi and rxni (LOS DTCT bit of stsrxi is high), the clock recovery algorithms are locked. This ensures the frequency of rxclk[i] is preserved, and that re-acquisition of phase alignment when LOS DTCT goes low occurs in the minimal time.

### 9.2.5.1 First Order Clock Recovery Algorithm

Whenever the difference between early and late votes equals or exceeds the specified threshold, the sampling position is adjusted by 1 UI. The results from the subsequent three voting windows are then ignored to allow the control loop to settle.

If the threshold is set to 1, this results in a maximum tracking rate of 1/64 UI per 32 bits,  $\pm 488$  ppm ( $1/(64 \times 32)$ ). If the threshold is set to 17, this results in a maximum tracking rate of  $\pm 325$  ppm ( $1/(64 \times 48)$ ). Actual tracking rate will typically be lower than this. How much will depend on the transition density of the data. In general, the tracking rate can be expressed mathematically as shown in equation 3, below, where T is the threshold and D is the transition density. The multiplicand in the denominator is rounded up to the nearest 8 UI, as this is the size of the voting window.

**Equation 3. 1<sup>st</sup> Order CDR Tracking Rate**

$$\text{Tracking Rate} = \frac{1}{64 \times \text{RND}_8((T/D) + 24)} \text{ppm}$$

This can be used to consider both typical and worst case scenarios. For example:

- If T is 17, and the average transition density is 1 in 5, it will take  $17 \times 5 = 85$  UI to reach the threshold. So, average tracking rate would be

$$\frac{1}{64 \times \text{RND}(85 + 24)} = \frac{1}{64 \times 112} = 139 \text{ ppr}$$

- If T is 1, and the worst case transition density is 1 in 70, it will take 70 UI to reach the threshold, which rounds up to 72 UI. So, worst case tracking rate would be

$$\frac{1}{64 \times (72 + 24)} = 162 \text{ ppm}$$

Even so, this analysis is slightly optimistic, as it assumes there are no incorrect votes. These may occur in the presence of jitter, and may affect how long it takes to exceed the selected threshold. For a worst case analysis, any such optimism is generally offset by the fact that the worst case pattern (edges at minimum transition density occurring indefinitely with no higher density periods) is a pathological case that does not actually occur.

It is important to ensure that the time between votes being taken and the phase adjustment occurring is kept to a minimum. If possible, it should be kept within the 24 UI blanking interval in order to minimize voting jitter in the first order algorithm.

Worst case lock time requires the phase of the sampling clocks to be adjusted by 1 UI. How long this takes will depend on the amount of underlying frequency offset that needs to be tracked. This is expressed mathematically in equation 4 (where Tracking rate is defined by “[Equation 3. 1st Order CDR Tracking Rate](#)” on page 43).

**Equation 4. Lock time (UI)**

$$T_{\text{LOCK}} (\text{rxpirxni}) = \frac{5 \times 10^5}{\text{Tracking Rate} - \text{frequency offset (ppm)}} \text{ UI}$$

### 9.2.5.2 Second Order Clock Recovery Algorithm

The second order algorithm keeps a running total of the net number of times the selected first order threshold is equalled or exceeded in a signed accumulator. An excess of late votes increment the accumulator, and an excess of early votes decrement it. If the accumulator is positive, regular positive phase adjustments will be made, and if negative, regular negative phase adjustments will be made. The larger the magnitude of the accumulator, the more frequent the phase adjustments will be. When the number of regular adjustments being made is enough to compensate for the underlying frequency offset, the number of early and late vote excesses will balance on average, and the accumulator value will stabilize at a value proportional to the frequency offset. Short term phase changes will be tracked by the first order algorithm, which continues to operate as normal.

The maximum and minimum interval between phase adjustments is selected via CDR, which determines the number of accumulator bits (8 or 10). If the accumulator contains N bits (excluding the sign bit), then the maximum interval between adjustments is one adjustment every  $2^{N+1}$  bits. This determines the precision with which frequency adjustments can be made ( $1 / (24 \times 2^{N+1})$  ppm). The minimum

interval between adjustments is theoretically four times every 8 bits, which occurs when the accumulator is at its maximum positive or negative value. However, to avoid excessively large lock times, the maximum allowable accumulator value is limited to allow no more than 3 or 1 adjustments every 8 bits ( $\pm 5860$  ppm or  $\pm 1953$  ppm respectively). This is summarized in [Table 18](#).

**Table 18 Second Order Clock Recovery Algorithm Parameters**

CDR	Frequency Offset (ppm)	Accumulator (bits)	Precision (ppm)	df/dt (ppm/UI)
0 1 0	1953	10	7.6	0.24
0 1 1	1953	10	7.6	0.07
1 0 0	5860	8	30.5	0.95
1 0 1	5860	8	30.5	0.28

The higher the precision, the more accurately the frequency offset can be matched, but the longer it takes to lock onto the frequency offset or respond to a change in frequency offset. Conversely, a lower precision will result in less accurate frequency offset matching, but faster lock times and quicker response to a change in frequency offset. Lock and response time is also affected by the threshold selected for the first order algorithm. Which setting to use will depend on the application. If the frequency offset is fixed, or varies very slowly, a high precision setting should be used. However, if the frequency offset varies (as it does in some applications that use spread spectrum clocking), the lower precision must be selected in order to be able to adequately track the change in frequency offset. Note that with low precision setting, there is a small probability, in the event of some form of unexpected soft error, that the CDR could lock on to a harmonic of the actual frequency offset if that offset is always less than  $\pm 4000$  ppm.

As with the first order loop, whenever a majority vote occurs, the results from the subsequent three voting windows are usually ignored to allow the control loop to settle. This determines the rate of change of frequency offset, and in turn the lock time. In this case, the subsequent three voting windows are not ignored provided there were at least 2 votes, and the vote was unanimous.

[Table 18](#) shows the precision and frequency tracking rates available. Tracking rates (df/dt) quoted are the maximum possible, and will reduce as a function of transition density. In general, the rate of change of frequency achievable from the second order CDR loop is the ppm rate of the first order loop, divided by  $2N+1$  (where N is the accumulator size), as shown in equation 5.

**Equation 5. 2<sup>nd</sup> Order CDR Tracking Rate**

$$\text{Tracking Rate} = \frac{1}{64 \times \text{RND}((T/D) + 24) \times 2^{N+1}} \text{ ppm}$$

### 9.2.6 Jitter Tolerance

The incoming data stream may contain several forms of jitter, which must be filtered by the receiver to ensure successful reception of the data. High frequency data dependent jitter caused by channel insertion loss will be compensated for by the adaptive equalizer, which precedes all other receive circuitry in the data path.

High frequency jitter that exhibits a probability distribution with a zero mean will be rejected by the first order clock recovery algorithm, which applies an averaging technique to detected edge positions to determine the ideal data sample point.

Low frequency jitter, which may be sinusoidal or due to constant frequency error, will also be filtered by the clock recovery algorithm tracking function and limited only by the maximum tracking rate. These different components of jitter in the receive eye should be assessed individually to determine the overall jitter tolerance.

### 9.2.7 Serial to Parallel conversion

The receiver bus bandwidth is fixed in hardware to 10 bits and is not configurable through an MMR.

### 9.2.8 Symbol Alignment

Each receiver independently supports two forms of symbol alignment, selectable via the ALIGN field, as shown in [Table 19](#)

**Table 19 Receiver Symbol Alignment Selection**

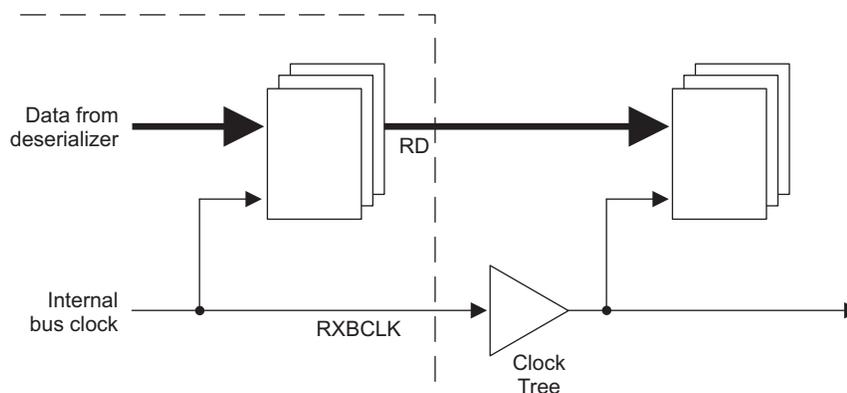
ALIGN	Effect
00	Alignment Disabled. No symbol alignment will be performed whilst this setting is selected, or when switching to this selection from another.
01	Comma alignment enabled. Symbol alignment will be performed whenever a misaligned comma symbol is received.
10	Alignment Jog. The symbol alignment will be adjusted by one bit position when this mode is selected.
11	Reserved

#### 9.2.8.1 Comma-Based Alignment

Setting the ALIGN field to 01 enables alignment to the K28 comma symbols included in the 8b:10b data encoding scheme defined by the IEEE and employed by numerous transmission standards.

In this mode, the receiver examines the data stream for the positive or negative disparity comma characters (001111xxx and 1100000xxx respectively). If a comma that straddles symbol boundaries is detected, the alignment will be corrected by adjusting the relationship between rxbclk[i] and rdi[19:0] appropriately. This will result in one symbol of corrupted data, as between 1 and 9 bits are removed from the data stream.

**Figure 5 rcbclk[i] and rdi[19:0] Interface**



Furthermore, if the 3 bits preceding the comma are the same as the MSB of the comma, it will be ignored. This prevents incorrect re-alignment happening in applications that use K28.7 characters. (A K28.7 followed by certain other symbols can produce what looks like a comma pattern across the symbol boundary. The K28.7 will be aligned to, but not the spurious comma that follows.)

Two mechanisms are used to perform the realignment:

- The period of `rxbclk[i]` is lengthened 2, 4, 6 or 8 bit periods. This is done in such a way that the clock is never high or low for less than the regular minimum.
- If the symbol is misaligned by an odd number of bits, a further single bit adjustment is made in the serial to parallel conversion. The deserializer contains 11 bits, and the appropriate 10-bits are mapped to `rdi[19:0]` using a remainder correction multiplexer.

When correcting a misalignment that is not a multiple of 2 bits, the number of bit periods that `rxbclk[i]` is stretched depends on the initial position of the remainder correction multiplexer. For example, a 3-bit misalignment could be corrected by a 2 bit period stretch of `rxbclk[i]`, and one bit more in the deserializer, or by a 4 bit period stretch and one bit less in the deserializer.

Whilst comma detection is enabled, the SYNC bit of RX status register `stsrx` will be driven high synchronously to `rxbclk[i]` when a comma is output on `rdi[19:0]`, provided that no realignment was required (i.e. the comma was already correctly aligned).

Comma re-alignment is not intended for use in systems that do not contain comma symbols (for example, unencoded 8-bit data symbols). As such, comma alignment is performed only if enabled when `BUSWIDTH = 000`.

Note that the entire family of K28 symbols are acted upon (i.e. the 3 LSBs are ignored).

When comma detection is disabled, the present alignment will be maintained, and SYNC will remain low. However, as a result of internal pipelining, it is possible for a comma received just before disabling to cause either a realignment or SYNC to activate.

#### 9.2.8.2 Single Bit Alignment Jog

For systems that cannot use comma-based symbol alignment, the single bit alignment jog capability provides a means to control the symbol realignment features of the receiver directly from logic implemented in the ASIC core. This logic can be designed to support whatever alignment detection protocol is required.

When a rising edge is detected on the MSB of ALIGN, a single bit adjustment to the alignment will be made (one data bit will be lost), in exactly the same way as for a comma misaligned by one bit

### 9.2.9 Loss of Signal Detection

Each receive channel also supports loss of signal detection, configured via the LOS bits of `cfgrx` register. The `LOSi` bits should be set to 2b10 to allow loss of signal detection by the SerDes macro. For internal loopback, this should be disabled.

### 9.3 SerDes Transmitter Configuration

The Transmitter configuration within the PCIe SerDes interface is configured using these registers:

- The PCIe SerDes Configuration Lane 0 register (SERDES\_CFG0)
- The PCIe SerDes Configuration Lane 1 register (SERDES\_CFG1)

For details on PCIe SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture PCIe User Guide*, in “[Related Documentation From Texas Instruments](#)” on page 55.

The Transmitter configuration within the SGMII SerDes interface is configured using these registers:

- The SGMII Transmitter Configuration Register 0 (SGMII\_SERDES\_CFGTX0)
- The SGMII Transmitter Configuration Register 1 (SGMII\_SERDES\_CFGTX1)

For details on SGMII SerDes configuration registers, definitions, and settings, see the *KeyStone Architecture GbE User Guide*, in “[Related Documentation From Texas Instruments](#)” on page 55.



**Note**—The PCIe interface uses fixed default values for the SERDES\_CFG0 and SERDES\_CFG1 registers.

#### 9.3.1 Data Rate Selection

The primary operating frequency of the SerDes macro is determined by the reference clock frequency and PLL multiplication factor. However, to support lower frequency applications, each receiver can also be configured to operate at a half, quarter, or one-eighth of this rate via the RATE bits as described in [Table 20](#).

**Table 20 Receiver Operating Rate**

Value	Rate
00	Full rate. Four data samples taken per PLL output clock cycle.
01	Half rate. Two data sample taken per PLL output clock cycle.
10	Quarter rate. One data sample taken per PLL output clock cycle.
11	Eighth rate. One data sample taken every two PLL output clock cycles.



**Note**—SGMII interface supports only the half rate mode.

#### 9.3.2 Serial to Parallel Conversion

The receiver bus bandwidth is fixed in hardware to 10 bits and is not configurable through an MMR.

#### 9.3.3 Differential Voltage-Mode Driver

The differential voltage-mode driver takes each bit from the parallel to serial converter and drives txpi and txni outputs accordingly. The polarity of txpi and txni can be inverted by setting the INVPAIR bit of the cfgtxi registers.

### 9.3.4 Output Voltage Swing Control

The output swing of the differential driver of each transmitter can be independently set to one of a number of settings via the SWING bits of `cfgtxi` register, as described in [Table 21](#). Reducing the output amplitude decreases the current drawn from VDDR in direct proportion to the reduction in swing, thereby saving power.

**Table 21 Differential Output Swing**

Value	DC-coupled Amplitude (mV <sub>dfpp</sub> )	AC-coupled Amplitude (mV <sub>dfpp</sub> )
0000	110	120
0001	190	200
0010	270	280
0011	350	360
0100	430	440
0101	510	530
0110	590	610
0111	670	690
1000	750	770
1001	840	850
1010	930	920
1011	1000	1010
1100	1080	1090
1101	1160	1170
1110	1250	1230
1111	1310	1330
<b>End of Table 1-21</b>		

The standard specifies several compliant interconnect and associated far-end eye templates. The swing and equalization settings required depends on which of these templates is being targeted by the application. If the Equalization function is not used, these setting should be 0000 for all.

### 9.3.5 Output Common Mode Adjustment

Always write 1 to this register field to select AC-coupled operation. All other values are reserved.

### 9.3.6 De-emphasis

De-emphasis provides a means to compensate for high frequency attenuation in the attached media. It causes the output amplitude to be smaller for bits that are not preceded by a transition than for bits. Sixteen different de-emphasis settings are provided via the DE bits of `cfgtxi`, as described in [Table 22](#). Note that the de-emphasis values in this table are defined to be the reduction in amplitude of a bit with the same polarity as its two or more immediate predecessors, with respect to the amplitude of a bit with opposite polarity to its immediate predecessor.

**Table 22 Differential Output De-emphasis (Part 1 of 2)**

Value	Amplitude Reduction (%)	Amplitude Reduction (dB)
0000	0	0
0001	4.76	-0.42
0010	9.52	-0.87

**Table 22 Differential Output De-emphasis (Part 2 of 2)**

Value	Amplitude Reduction (%)	Amplitude Reduction (dB)
0011	14.28	-1.34
0100	19.04	-1.83
0101	23.8	-2.36
0110	28.56	-2.36
0111	33.32	-2.92
1000	38.08	-4.16
1001	42.85	-4.86
1010	47.61	-5.61
1011	52.38	-6.44
1100	57.14	-7.35
1101	61.9	-8.38
1110	66.66	-9.54
1111	71.42	-10.87
<b>End of Table 1-22</b>		

## 10 SerDes Lane-Setting Optimization Procedures

The following sections describe system-level procedures that can be used to optimize the PLL, TX, and RX performance of all SerDes lanes.

### 10.1 Optimizing PLL Performance for Reducing TX Jitter and Increasing RX CDR Margin

The PLL VCO operating frequency must be kept well within the center of its operating range to minimize TX jitter and maximize RX CDR robustness. This can be done by:

- Minimizing PLL multiplier value
- Maximizing lane-rate
- Maximizing reference clock frequency

It may not be possible to adjust many of these parameters for a given interface if the industry specification mandates standard reference clock frequencies and lane-rates

### 10.2 Optimizing Transmitter SWING and FIR Filter Settings for Optimal Power and Receiver Performance

This section describes procedures that can be used to optimize the SerDes transmitter swing and FIR filter settings. The goal is to achieve maximum margin at the intended receiver over a given PCB channel. These methods are applicable to both the PCB simulation environment and final PCB platform. All of the transmitters for the SerDes-based peripherals have programmable transmitter output voltage swing that can be adjusted independently for each lane. The AIF2, SRIO, and Hyperlink transmitters also include a 3-tap FIR filter that can be adjusted independently for each lane. The SerDes transmitters for the SGMII and PCIe peripherals include a 1-tap, de-emphasis filter that can be adjusted independently for each lane.

Each SerDes-based peripheral will have different transmitter configuration limitations and caveats. See the peripheral-specific user guide for a full list of all available settings that are accessible through the SerDes configuration register interface. These settings must be optimized together for each lane of the PCB system to maximize the signal eye width and height and ensure that the signal is not over or under-equalized at the intended receiver. Attenuation of a signal through a PCB transmission line tends to increase as the frequency of the signal increases such that most channels act like a low-pass filter. Given these losses, the transmitter swing must be increased as the transmission line losses increase to provide uniform gain of the signal.

However, increased swing can also result in an over-equalized signal at the receiver, increased cross-talk between adjacent signals and increased power consumption in the system. Industry standard receiver eye-diagram inner-eye and out-eye minimum and maximum constraints help to bound this condition at the receiver interface to the PCB. Given this situation, there are a few simultaneous constraints placed on transmitter swing amplitude settings:

- Higher transmitter swing amplitudes can contribute to larger cross-talk voltage magnitudes, over-equalization at the receiver and increased power consumption
- Lower transmitter swing amplitudes can result in closed eye at the receiver

The FIR filter coefficients should be set such that as much of the frequency dependent losses are pre-equalized before entering the transmission-line. However, care must be taken not to over or under-equalize the signal with the FIR. Under-equalization can result in a situation where the receiver adaptive equalizer may not be able to recover the signal after the channel losses are applied. Likewise, over-equalization can saturate the receiver sampler, which can lead to sampler errors.

Industry standard receiver eye-diagram minimum and maximum constraints bound this condition at the receiver interface to the PCB.

Given the effects of the FIR filter on the signal, there are a few simultaneous constraints placed on FIR filter coefficients settings:

- Certain FIR coefficients can result in an over-equalized, saturated receiver samplers
- Certain FIR coefficients can result in an under-equalized, closed, receiver eye diagram

### 10.2.1 Creating Initial Values for Swing and FIR Settings

Simulation of the target PCB platform with appropriate 3D EM modeling and the KeyStone I IBIS-AMI SerDes models is the best method of creating initial starting values for the actual PCB.



**Note**—To obtain more details on KeyStone I IBIS-AMI models, please contact the TI sales team.

However, the vast majority of SerDes channels in KeyStone I SoC PCB systems are short-run KeyStone I-to-KeyStone I transmission lines. These short-run channels tend to have insertion losses in the 5 dB to 10 dB range (or less) at the operating frequency and edge-rates the KeyStone I SerDes peripherals operate at.

With these short-run channels, maximizing the transmitter swing value while enabling the receiver fully adaptive equalizer usually results in an operational link. A small amount of negative FIR filter post-cursor tap (-2.5% or -5.0%) is enough to usually pre-equalize the transmit signal. In the case of longer channels and those with interconnects and cabling, the best method for creating initial values is full 3D EM modeling of the system.

### 10.2.2 Creating Valid Swing and FIR Settings Map Based on Receiver Errors Detected

The simplest method for creating valid transmitter settings is to implement an exhaustive search across the entire transmitter swing and FIR or de-emphasis settings space for each link partner while leaving the link receivers in fully-adaptive equalizer mode (or an equivalent receiver equalizer mode for non-KeyStone I link partners).

A test program can be setup that sweeps through each transmitter setting while live traffic is exchanged between link partners. The test programs creates and stores error rate maps based on the ratio of the number of bits received with no errors compared to those with errors over a given window of time for a given transmitter setting iteration. Case temperature of the SoC and other system-level environmental conditions can also be simultaneously swept to create a multi-dimensional map of the entire operational space with respect to SerDesoperation.

This error-rate map can then be analyzed offline in any convenient data analysis tool. Many methods can be used to determine the exact parameters that will result in the most operating margin of the SerDes link such as linear regressions, least-squares maximization etc. This procedure should be done for all SerDes lanes on a given platform as even small placement, routing and environmental changes between lanes can result in different optimal settings for a given lane. The same procedure can also be used to analyze compliance with industry standard receiver eye diagrams and jitter specifications.

## 11 Terminations

All SerDes-based interfaces should be AC-coupled. As long as the SerDes link partner uses CML logic, the AC-coupling capacitor is the only external termination required. For AC-coupling, the recommendation is to use an 0402 or smaller 0.1- $\mu$ F ceramic capacitor placed closest to the receiver BGA for all SerDes interfaces except PCIe. For PCIe the AC-coupling capacitor should be placed closest to the transmitter BGA. This should be the case for the Serial RapidIO because that standard calls for CML signals. The SGMII specification calls for low-voltage differential signaling (LVDS), so additional terminations may be required. The need for terminations is dependent on the internal terminations in the link partner device.

See *Clocking Design Guide for KeyStone Devices* for recommendations on connecting two different clocking techniques together, in [“Related Documentation From Texas Instruments”](#) on page 55.

## 12 Related Documentation From Texas Instruments

<a href="#">AIF1-to-AIF2 Antenna Interface Migration Guide for KeyStone Devices</a>	SPRABH8
<a href="#">Antenna Interface 2 (AIF2) for KeyStone Devices User Guide</a>	SPRUGV7
<a href="#">Clocking Design Guide for KeyStone Devices</a>	SPRABI4
<a href="#">Connecting AIF2 with FFTC</a>	SPRABF3
<a href="#">DDR3 Design Guide for KeyStone Devices</a>	SPRABI1
<a href="#">General Purpose AIF2 Traffic for KeyStone Devices</a>	SPRABH3
<a href="#">Gigabit Ethernet (GbE) Switch Subsystem for KeyStone Devices User Guide</a>	SPRUGV9
<a href="#">Hardware Design Guide for KeyStone Devices</a>	SPRABI2
<a href="#">HyperLink for KeyStone Devices User Guide</a>	SPRUGW8
<a href="#">HyperLink Use Cases for KeyStone Devices</a>	SPRABH5
<a href="#">Layer 2 (PA/SA/MultiCore Navigator) Applications for KeyStone Devices</a>	SPRABH7
<a href="#">Network Coprocessor (NETCP) for KeyStone Devices User Guide</a>	SPRUGZ6
<a href="#">Packet Accelerator (PA) for KeyStone Devices User Guide</a>	SPRUGS4
<a href="#">Peripheral Component Interconnect Express (PCIe) for KeyStone Devices User Guide</a>	SPRUGS6
<a href="#">Phase Locked Loop (PLL) Controller for KeyStone Devices User Guide</a>	SPRUGV2
<a href="#">Serial RapidIO (SRIO) for KeyStone Devices User Guide</a>	SPRUGW1
<a href="#">SRIO Migration Guide (F to N) for KeyStone Devices</a>	SPRABI0
<a href="#">SRIO Usage Considerations for KeyStone Devices</a>	SPRABH4
<a href="#">TMS320C6608 Multicore Fixed and Floating-Point Digital Signal Processor Data Manual</a>	SPRS623
<a href="#">TMS320TCI6608 Multicore Fixed and Floating Point DSP Silicon Errata</a>	SPRZ333
<a href="#">TMS320C6670 Multicore Fixed and Floating-Point System-on-Chip Data Manual</a>	SPRS689
<a href="#">TMS320C6670 Multicore Fixed and Floating-Point System-on-Chip Silicon Errata</a>	SPRZ332
<a href="#">TMS320C6672 Multicore Fixed and Floating-Point Digital Signal Processor Data Manual</a>	SPRS708
<a href="#">TMS320C6672 Fixed and Floating-Point Digital Signal Processor Silicon Errata</a>	SPRZ335
<a href="#">TMS320C6674 Multicore Fixed and Floating-Point Digital Signal Processor Data Manual</a>	SPRS692
<a href="#">TMS320C6674 Fixed and Floating-Point Digital Signal Processor Silicon Errata</a>	SPRZ341
<a href="#">TMS320C6678 Multicore Fixed and Floating-Point Digital Signal Processor Data Manual</a>	SPRS691
<a href="#">TMS320C6678 Fixed and Floating-Point Digital Signal Processor Silicon Errata</a>	SPRZ334
<a href="#">TMS320TCI6614 Communications Infrastructure KeyStone System-on-Chip Data Manual</a>	SPRS671
<a href="#">TMS320TCI6616 Communications Infrastructure KeyStone System-on-Chip Data Manual</a>	SPRS624
<a href="#">TMS320TCI6616 Communications Infrastructure KeyStone SOC Silicon Errata</a>	SPRZ331
<a href="#">TMS320TCI6618 Communications Infrastructure KeyStone System-on-Chip Data Manual</a>	SPRS688

## 13 Revision History

The following table lists changes for each revision.

<b>Revision</b>	<b>Date</b>	<b>Description</b>
SPRABC1	October 2012	First Release

## 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 which meet ISO/TS16949 requirements, mainly for automotive use. Components which have not been so designated are neither designed nor intended for automotive use; and TI will not be responsible for any failure of such components to meet such requirements.

### Products

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Applications Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Automotive and Transportation	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

### TI E2E Community

[e2e.ti.com](http://e2e.ti.com)