SDI IP Core User Guide

Updated for Intel® Quartus® Prime Design Suite: **19.1**

Online Version
Send Feedback

UG-SDI1005

ID: 683587
Version: 2020.08.20

Contents

1. SDI IP Core Overview

The Serial Digital Interface (SDI) IP core implements a receiver, transmitter, or full-duplex SDI at standard definition (SD), high definition (HD), or 3 gigabits per second (3G). The SDI IP core supports dual standard (HD-SDI and SD-SDI) and triple standard (SD-SDI, HD-SDI, and 3G-SDI). These modes provide automatic receiver rate detection.

Attention: Intel has discontinued support for the SDI IP core. You may continue using this IP without support. However, Intel recommends that you use the SDI II Intel® FPGA IP for the latest features and continued support.

Table 1. SDI IP Core Features

Feature	Description
Support	• Multiple SDI standards and video formats • RP168 video switch line requirement
Transmitter	• Cyclical redundancy check (CRC) encoding (HD only) • Line number (LN) insertion (HD only) • Word scrambling • Transmitter clock multiplexer
Receiver	• CRC decoding (HD only) • LN extraction (HD only) • Framing and extraction of video timing signals • Word alignment and descrambling
IP Catalog	Easy-to-use parameter editor

Related Information

- SDI II Intel FPGA IP User Guide
- SDI Audio Intel FPGA IP User Guide

1.1. Release Information

The following table lists information about this release of the SDI IP core.

Table 2. Release Information

Item	Description
Version	19.1
Release Date	September 2019
Ordering Code	IP-SDI
Product ID	00AE
Vendor ID	6AF7

Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

Intel verifies that the current version of the Intel Quartus® Prime software compiles the previous version of each IP core, if this IP core was included in the previous release. Any exceptions to this verification are reported in the *Serial Digital Interface (SDI) IP Core Release Notes*. Intel does not verify compilation with IP core versions older than the previous release.

Related Information

- Errata for SDI IP core in the Knowledge Base
- Serial Digital Interface (SDI) IP Core Release Notes

1.2. Device Family Support

The table below lists the device support information for the SDI IP core.

Table 3. Device Family Support

Device Family	Support
Arria® II GZ	Final
Arria V	Final
Cyclone® IV GX (1)	Final
Cyclone V	Final
Stratix® IV (2)	Final
Stratix V (2)	Final
Other device families	No support

1.3. General Description

The Society of Motion Picture and Television Engineers (SMPTE) defines an SDI standard that video system designers use widely as an interconnect between equipment in video production facilities.

The SDI IP core handles the following SDI data rates:

- 270 megabits per second (Mbps) SD-SDI, as defined by *SMPTE259M-1997 10-Bit 4:2:2 Component Serial Digital Interface*
- 1.5 gigabits per second (Gbps) HD-SDI, as defined by *SMPTE292M-1998 Bit-Serial Digital Interface for High Definition Television Systems*
- 3-Gbps SDI, as defined by *SMPTE425M-AB 2006 3Gb/s Signal/Data Serial Interface– Source Image Format Mapping*
- Preliminary dual link SDI support, as defined by *SMPTE372M-Dual Link 1.5Gb/s Digital Interface for 1920×1080 and 2048×1080 Picture Formats*
- Dual standard support for 270-Mbps and 1.5-Gbps SDI

(1) Transceiver dynamic configuration with channel reconfiguration mode is not supported for dual and triple standard in EP4CGX110 and EP4CGX150 devices. Use Final transceiver dynamic reconfiguration with PLL reconfiguration mode instead.

(2) If you have only 27 MHz to drive the SDI IP core in SD-SDI mode, you require an additional PLL to generate a 67.5-MHz reference clock.

- Triple standard support for 270-Mbps, 1.5-Gbps, and 3-Gbps SDI
- Triple standard support for SD-SDI, HD-SDI, and 3G-SDI
- *SMPTE425M* Level A support (direct source image formatting)
- *SMPTE425M* Level B support (dual link mapping)

Table 4. SDI Standard Support

All SDI standards, except SD-SDI, require a transceiver-based device.
Table below lists the SDI standard support for various FPGA devices.

Device Family	SDI Standard					
	SD-SDI	HD-SDI	3G-SDI	HD-SDI Dual Link(3)	Dual Standard	Triple Standard
Arria II GX	Yes	Yes	Yes	Yes	Yes	Yes
Arria V	Yes	Yes	Yes	Yes	Yes	Yes
Stratix IV (4)	Yes	Yes	Yes	Yes	Yes	Yes
Stratix V (4)	Yes	Yes	Yes	Yes	Yes	Yes
Cyclone IV GX (EP4CGX15, EP4CGX30)	Yes	—	—	—	—	—
Cyclone IV GX (EP4CGX30 (F484), EP4CGX50, EP4CGX75, EP4CGX110, EP4CGX150)	Yes	Yes	Yes	Yes	Yes	Yes
Cyclone V (5)	Yes	Yes	Yes	Yes	Yes	Yes

1.4. Resource Utilization

The table below lists the typical resource utilization for various parameters of the SDI IP core with the Intel Quartus Prime Standard Edition software.

Note: The resource utilization of the IP core is based on the bidirectional interface settings unless otherwise specified.

Table 5. Resource Utilization

Device	Video Standard	LEs	Combinational ALUTs	Logic Registers
Arria II GX	SD-SDI	—	839	680
	HD-SDI	—	978	833

continued...

(3) The HD-SDI dual link supports timing difference up to 40 ns between link A and link B, fulfilling the SMPTE372M requirement.

(4) Only Stratix IV and Stratix V variants with transceivers support all SDI rates.

(5) The 3G-SDI standard is not supported in Cyclone V devices with transceiver speed grade 7, due to the excessive data rate required.

Device	Video Standard	LEs	Combinational ALUTs	Logic Registers
	3G HD-SDI	—	1,259	1,015
	Dual-Link HD-SDI	—	2,029	1,711
	Dual standard receiver	—	1,257	926
	Dual standard transmitter	—	267	180
	Triple standard	—	1,891	1,305
Arria V	SD-SDI	—	1,189	920
	HD-SDI	—	1,185	910
	3G HD-SDI	—	1,444	1,142
	Dual-Link HD-SDI	—	2,446	1,880
	Dual standard receiver	—	1,605	1,175
	Dual standard transmitter	—	349	269
	Triple standard	—	2,273	1,677
Cyclone IV GX (EP4CGX15, EP4CGX30)	SD-SDI	916		
Cyclone IV GX (EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150)	SD-SDI	—	1,129	671
	HD-SDI	—	1,164	670
	3G HD-SDI	—	1,409	790
	Dual-Link HD-SDI	—	2,515	1,467
	Dual standard receiver	—	1,479	755
	Dual standard transmitter	—	364	229
	Triple standard	—	2,235	1,121
Cyclone V	SD-SDI	—	1,140	832
	HD-SDI	—	1,122	808
	3G HD-SDI	—	1,402	997
	Dual-Link HD-SDI	—	2,351	1,696
	Dual standard receiver	—	1,539	1,042
	Dual standard transmitter	—	352	260
	Triple standard	—	2,217	1,508
Stratix IV	SD-SDI	—	839	680
	HD-SDI	—	978	833
	3G HD-SDI	—	1,259	1,015
	Dual-Link HD-SDI	—	2,029	1,711
	Dual standard receiver	—	1,257	926

continued...

Device	Video Standard	LEs	Combinational ALUTs	Logic Registers
	Dual standard transmitter	—	267	180
	Triple standard	—	1,891	1,305
Stratix V	SD-SDI	—	913	707
	HD-SDI	—	955	703
	3G HD-SDI	—	1,126	823
	Dual-Link HD-SDI	—	2,049	1,522

2. SDI IP Core Getting Started

2.1. Installing and Licensing Intel FPGA IP Cores

The Intel Quartus Prime software installation includes the Intel FPGA IP library. This library provides many useful IP cores for your production use without the need for an additional license. Some Intel FPGA IP cores require purchase of a separate license for production use. The Intel FPGA IP Evaluation Mode allows you to evaluate these licensed Intel FPGA IP cores in simulation and hardware, before deciding to purchase a full production IP core license. You only need to purchase a full production license for licensed Intel IP cores after you complete hardware testing and are ready to use the IP in production.

The Intel Quartus Prime software installs IP cores in the following locations by default:

Table 6. IP Core Installation Locations

Location	Software	Platform
`<drive>:\intelFPGA\quartus\ip\altera`	Intel Quartus Prime Standard Edition	Windows
`<home directory>:/intelFPGA/quartus/ip/altera`	Intel Quartus Prime Standard Edition	Linux

Note: The Intel Quartus Prime software does not support spaces in the installation path.

2.1.1. Intel FPGA IP Evaluation Mode

The free Intel FPGA IP Evaluation Mode allows you to evaluate licensed Intel FPGA IP cores in simulation and hardware before purchase. Intel FPGA IP Evaluation Mode supports the following evaluations without additional license:

- Simulate the behavior of a licensed Intel FPGA IP core in your system.
- Verify the functionality, size, and speed of the IP core quickly and easily.
- Generate time-limited device programming files for designs that include IP cores.
- Program a device with your IP core and verify your design in hardware.

Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

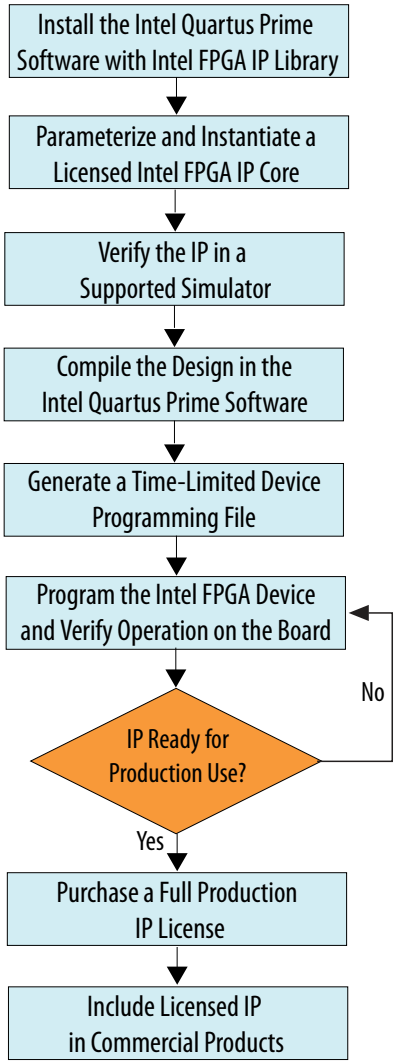
Intel FPGA IP Evaluation Mode supports the following operation modes:

- **Tethered**—Allows running the design containing the licensed Intel FPGA IP indefinitely with a connection between your board and the host computer. Tethered mode requires a serial joint test action group (JTAG) cable connected between the JTAG port on your board and the host computer, which is running the Intel Quartus Prime Programmer for the duration of the hardware evaluation period. The Programmer only requires a minimum installation of the Intel Quartus Prime software, and requires no Intel Quartus Prime license. The host computer controls the evaluation time by sending a periodic signal to the device via the JTAG port. If all licensed IP cores in the design support tethered mode, the evaluation time runs until any IP core evaluation expires. If all of the IP cores support unlimited evaluation time, the device does not time-out.
- **Untethered**—Allows running the design containing the licensed IP for a limited time. The IP core reverts to untethered mode if the device disconnects from the host computer running the Intel Quartus Prime software. The IP core also reverts to untethered mode if any other licensed IP core in the design does not support tethered mode.

When the evaluation time expires for any licensed Intel FPGA IP in the design, the design stops functioning. All IP cores that use the Intel FPGA IP Evaluation Mode time out simultaneously when any IP core in the design times out. When the evaluation time expires, you must reprogram the FPGA device before continuing hardware verification. To extend use of the IP core for production, purchase a full production license for the IP core.

You must purchase the license and generate a full production license key before you can generate an unrestricted device programming file. During Intel FPGA IP Evaluation Mode, the Compiler only generates a time-limited device programming file (*<project name>*`_time_limited.sof`) that expires at the time limit.

Figure 1. Intel FPGA IP Evaluation Mode Flow



Note: Refer to each IP core's user guide for parameterization steps and implementation details.

Intel licenses IP cores on a per-seat, perpetual basis. The license fee includes first-year maintenance and support. You must renew the maintenance contract to receive updates, bug fixes, and technical support beyond the first year. You must purchase a full production license for Intel FPGA IP cores that require a production license, before generating programming files that you may use for an unlimited time. During Intel FPGA IP Evaluation Mode, the Compiler only generates a time-limited device programming file (`<project name>_time_limited.sof`) that expires at the time limit. To obtain your production license keys, visit the Intel FPGA Self-Service Licensing Center.

The Intel FPGA Software License Agreements govern the installation and use of licensed IP cores, the Intel Quartus Prime design software, and all unlicensed IP cores.

Related Information

- Intel FPGA Licensing Support Center
- Introduction to Intel FPGA Software Installation and Licensing

2.2. IP Catalog and Parameter Editor

The IP Catalog displays the IP cores available for your project, including Intel FPGA IP and other IP that you add to the IP Catalog search path. Use the following features of the IP Catalog to locate and customize an IP core:

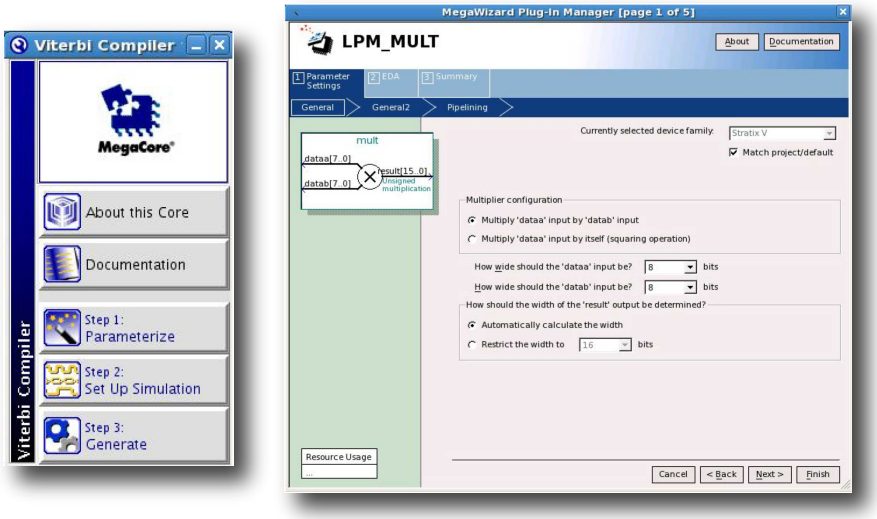
- Filter IP Catalog to **Show IP for active device family** or **Show IP for all device families**. If you have no project open, select the **Device Family** in IP Catalog.
- Type in the Search field to locate any full or partial IP core name in IP Catalog.
- Right-click an IP core name in IP Catalog to display details about supported devices, to open the IP core's installation folder, and for links to IP documentation.
- Click **Search for Partner IP** to access partner IP information on the web.

The parameter editor generates a top-level Quartus IP file (`.qip`) for an IP variation in Intel Quartus Prime Standard Edition projects. These files represent the IP variation in the project, and store parameterization information.

2.2.1. Generating IP Cores (Intel Quartus Prime Standard Edition)

This topic describes parameterizing and generating an IP variation using a legacy parameter editor in the Intel Quartus Prime Standard Edition software.

Figure 2. Legacy Parameter Editors



1. In the IP Catalog (**Tools ➤ IP Catalog**), locate and double-click the name of the IP core to customize. The parameter editor appears.
2. Specify a top-level name and output HDL file type for your IP variation. This name identifies the IP core variation files in your project. Click **OK**. Do not include spaces in IP variation names or paths.
3. Specify the parameters and options for your IP variation in the parameter editor. Refer to your IP core user guide for information about specific IP core parameters.
4. Click **Finish** or **Generate** (depending on the parameter editor version). The parameter editor generates the files for your IP variation according to your specifications. Click **Exit** if prompted when generation is complete. The parameter editor adds the top-level `.qip` file to the current project automatically.

Note: For devices released prior to Intel Arria® 10 devices, the generated `.qip` and `.sip` files must be added to your project to represent IP and Platform Designer systems. To manually add an IP variation generated with legacy parameter editor to a project, click **Project ➤ Add/Remove Files in Project** and add the IP variation `.qip` file.

2.2.2. Parameterizing the SDI IP Core

To parameterize your SDI IP core, follow these steps:

1. Select the video standard.
2. Select **Bidirectional**, **Transmitter**, or **Receiver** interface direction.
3. Click the **Transceiver Options** tab.
4. Under **Transceiver and Protocol**, click **Generate transceiver and protocol blocks**.
5. Select the starting channel number.
6. Turn on **Use PLL reconfiguration for transceiver dynamic reconfiguration** if you selected EP4CGX110 or EP4CGX150 device for Cyclone IV GX using dual and triple standards.

   Note: You may turn on this option for other Cyclone IV GX devices but it is not recommended.
7. Turn on **Enable TX PLL select for 1/1.000 and 1/1.001 data rate reconfiguration** if your design requires two serial input clocks to the TX block.

   Note: This feature is only available for the Arria II and Stratix IV GX device families.
8. Click the **Receiver/Transmitter Options** tab.
9. Turn on the necessary receiver options.
10. Turn on the necessary transmitter options.
11. Click **Next**.

Related Information

SDI IP Core Parameters on page 14

2.2.3. SDI IP Core Parameters

You can set the SDI parameters using the IP catalog.

Parameter		Value	Description
Protocol Options	Video standard	SD-SDI, HD-SDI, 3G-SDI, HD-SDI dual link, dual or triple standard SDI	Sets the video standard. • SD-SDI—includes oversampling logic. • HD-SDI—switches in LN insertion and extraction and CRC generation and extraction blocks; selecting SD-SDI switches out LN insertion and extraction and CRC generation and extraction blocks. • Dual or triple rate SDI—includes the processing blocks for both SD-SDI and HD-SDI standards. In addition, logic for bypass paths and logic to automatically switch between the input standards is included.
	Interface settings	Birectional, Receiver, Transmitter	Selects transceiver or protocol blocks or both. When non-GX device is chosen, only SD-SDI protocol block is permitted. If you want to generate HD-SDI or 3G-SDI protocol block, you must select a GX device.
Transceiver Option	Transceiver and Protocol	Generate transceiver and protocol blocks, generate transceiver block only, or generate protocol block only	Selects transceiver or protocol blocks or both. When non-GX device is chosen, only SD-SDI protocol block is permitted. If you want to generate HD-SDI or 3G-SDI protocol block, you must select a GX device.
	Use soft logic for transceiver	On or off	Uses soft logic to implement the transceiver logic, rather than using Stratix IV transceivers. SD-SDI only. For example, if you run out of hard transceivers in your device, you can implement the function in soft logic. If you have spare transceivers in a device, you may wish to use them.
	Starting channel number	0, 4, 8, ..., 156	Dual or triple standard only. Each dual or triple standard SDI must have a unique starting channel number. *Note:* This parameter is not applicable for Arria V, Cyclone V, and Stratix V devices.
	Use PLL reconfiguration for transceiver dynamic reconfiguration	On or off	Dual or triple standard, and Cyclone IV GX devices only. You must turn on this option if you select an EP4CGX110 or EP4CGX150 device.
	Enable TX PLL select for 1/1.000 and 1/1.001 data rate reconfiguration	On or off	Enables an additional input port for transmitter serial reference clock. *Note:* Available for Arria II and Stratix IV devices only.
Receiver/ Transmitter Options	CRC error output	On or off	• On: CRC monitoring (Not applicable for SD-SDI mode) • Off: No CRC monitoring (saves logic)
	SDI synchronization output	On or off	Provides synchronization outputs.
	Tolerance to consecutive missed EAV/SAV	0, 1, 2, ..., 15	Receiver protocol only. Allows you to set the number of consecutive missing EAVs to be tolerated in the incoming video. Specify a higher value if you want the receiver core to tolerate more errors.

continued...

Parameter		Value	Description
			If you want the receiver core not to tolerate any errors, set this option to 0.
	Two times oversample mode	On or off	HD-SDI transmitter only. When turned on, runs the transceiver at twice the rate and has improved jitter performance. Requires 148.5-MHz `tx_serial_refclk` reference clock.

3. Functional Description

The SDI IP core implements a transmitter, receiver, or full-duplex interface. The SDI IP core consists of the following components:

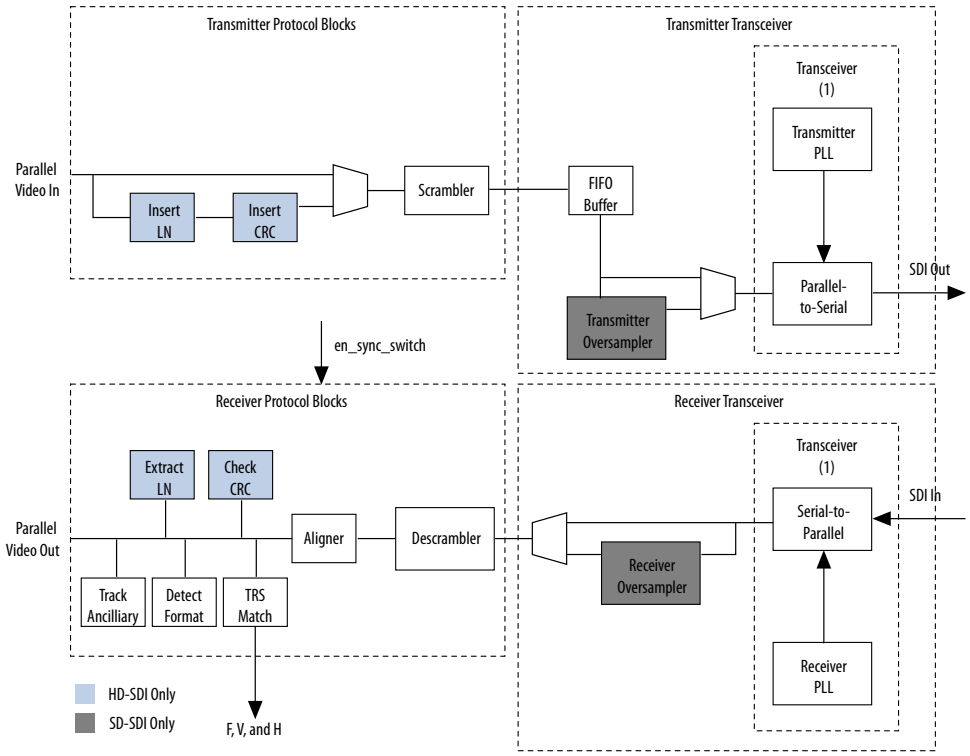
- Protocol block—transmitter or receiver
- Transceiver
- Transceiver controller

In the parameter editor, you can specify either protocol, transceiver, or combined blocks for your design. For example, if you have multiple protocol blocks in a design, you can multiplex the blocks into one transceiver.

The transceiver can be either a soft-logic implementation or a GX transceiver.

Figure 3. SDI IP Core Block Diagram

The figure shows the SDI IP core block diagram.



Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

3.1. Transmitter

The transmitter contains the following elements:

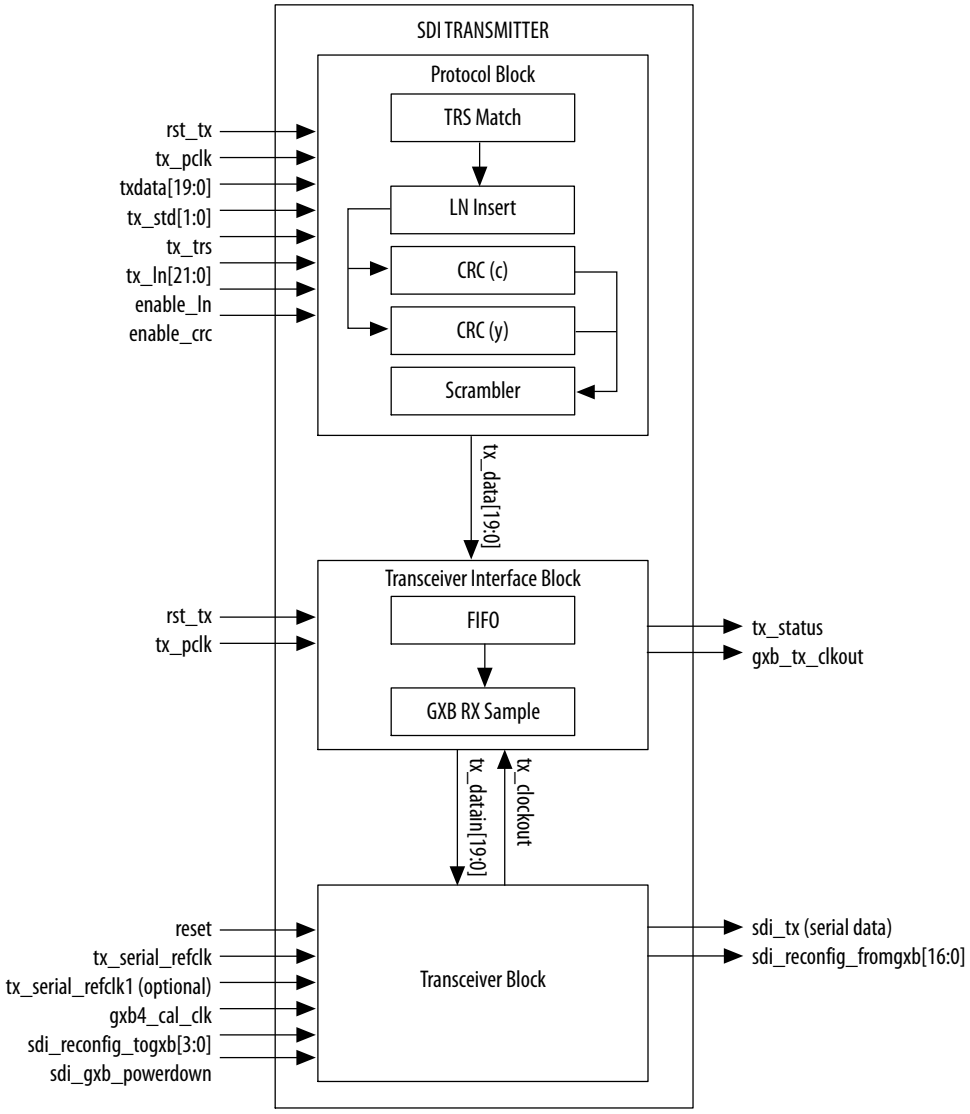
- SD/HD-SDI transmitter scrambler
- HD-SDI transmitter data formatter, which includes a CRC and LN insertion
- Transceiver, plus control, and interface logic with multirate (dual or triple standard) SD/HD-SDI transmitter operation
- Transmitter clock multiplexer (optional)

The transmitter performs the following functions:

- HD-SDI LN insertion
- HD-SDI CRC generation and insertion
- Clock enable signal generation
- Scrambling and non-return-zero inverted (NRZI) coding
- Internal switching between two reference clock signals in the transmitter block. This feature is optional and only available for Arria II GZ and Stratix IV GX

The figure shows the top-level block diagram for the SDI transmitter.

Figure 4. SDI Transmitter Block Diagram



For HD-SDI, the transmitter accepts 20-bit parallel video data; for SD-SDI, 10-bit parallel data.

Table 7. Bit Allocation for txdata for Supported Video Standards

The table below lists the bit allocation for `txdata`.

txdata	SD-SDI	HD-SDI	3G-SDI Level A	3G-SDI Level B
[19:10]	Unused	Y	Y	Cb, Y, Cr, Y multiplex (link A)
[9:0]	Cb, Y, Cr, Y multiplex	C	C	Cb, Y, Cr, Y multiplex (link B)

For HD-SDI operation, the current video line number is inserted at the appropriate point in each line. A CRC is also calculated and inserted for the luma and chroma channels.

The parallel video data is scrambled and NRZI encoded according to the SDI specification.

The transceiver converts the encoded parallel data into the high-speed serial output (parallel-to-serial conversion).

HD-SDI LN Insertion

SMPTE292M section 5.4 defines the format of two words that are included in each HD-SDI video line to indicate the current line number. The HD-SDI LN insertion module takes the lower 11-bit `tx_ln`, formats and inserts it as two words in the output data. The HD-SDI LN insertion module accepts the current line number as an input.

The LN words (LN0 and LN1) overwrite the two words that follow the "XYZ" word of the EAV TRS sequence. The same value is included in the luma and chroma channels. For correct LN insertion, you must assert the `tx_trs` signal must be asserted for the first word of both EAV and SAV TRSs (refer to Figure 3–31 on page 3–47 and Figure 3–32 on page 3–48).

Note: If the system does not know the line number, you can implement logic to detect the output video format and then determine the current line. This function is outside the scope of this SDI MegaCore function.

HD-SDI CRC Generation and Insertion

SMPTE292M section 5.5 defines a CRC that is included in the chroma and luma channels for each HD-SDI video line. The HD-SDI CRC module generates, formats, and inserts the required CRC in the output data.

The HD-SDI CRC module identifies the words that you must include in the CRC calculation, and also determines where you must insert the words in the output data. The formatted CRC data words (`YCR0` and `YCR1` for the luma channel, `CCR0` and `CCR1` for the chroma channel) overwrite the two words that follow the line number words after the EAV. The module provides aseparate calculation for the luma and chroma channels.

The module calculates CRC for all words in the active digital line, starting with the first active word line and finishing with the final word of the line number (LN1). The initial value of the CRC is set to zero, then the polynomial generator equation CRC(X) = X18 + X5 + X4 + 1 is applied.

The HD-SDI CRC module implements the CRC calculation by iteratively applying the polynomial generator equation to each bit of the output data, processing the LSB first.

For correct CRC generation and insertion, the `tx_trs` signal must be asserted for the first word of both EAV and SAV TRS (refer to Figure 3–31 on page 3–47 and Figure 3–32 on page 3–48).

Scrambling and NRZI Coding

SMPTE292M section 5 and SMPTE292M section 7 define a common channel coding that is used for both SDI and HD-SDI. This channel coding consists of a scrambling function (G1(X) = X9 + X4 + 1) followed by NRZI encoding (G2(X) = X + 1). The scrambling module implements this channel coding. You can configure the module to process either 10-bit or 20-bit parallel data.

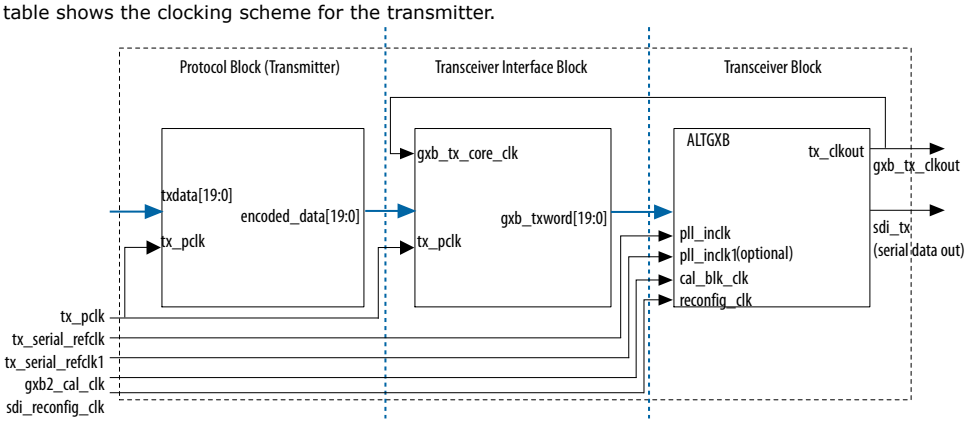
The scrambling module implements the channel coding by iteratively applying the scrambling and NRZI encoding algorithm to each bit of the output data, processing the LSB first. Figure C.1 of SMPTE259M shows how the algorithm is implemented.

Transceiver Clock

The `tx_serial_refclk1` is an optional port that is enabled when you turn on the **Enable TX PLL select for 1/1.000 and 1/1.001 data rate reconfiguration** in the parameter editor.

Figure 5. Transmitter Clocking Scheme

This table shows the clocking scheme for the transmitter.



3.2. Receiver

The receiver contains the following elements:

- SD/HD-SDI receiver descrambler and word aligner
- HD-SDI receiver CRC and LN extractor
- Transceiver, plus control, and interface logic with multirate (dual or triple standard) SD/HD-SDI transmitter operation
- Receiver framing, with extraction of video timing signals
- Identification and tracking of ancillary data

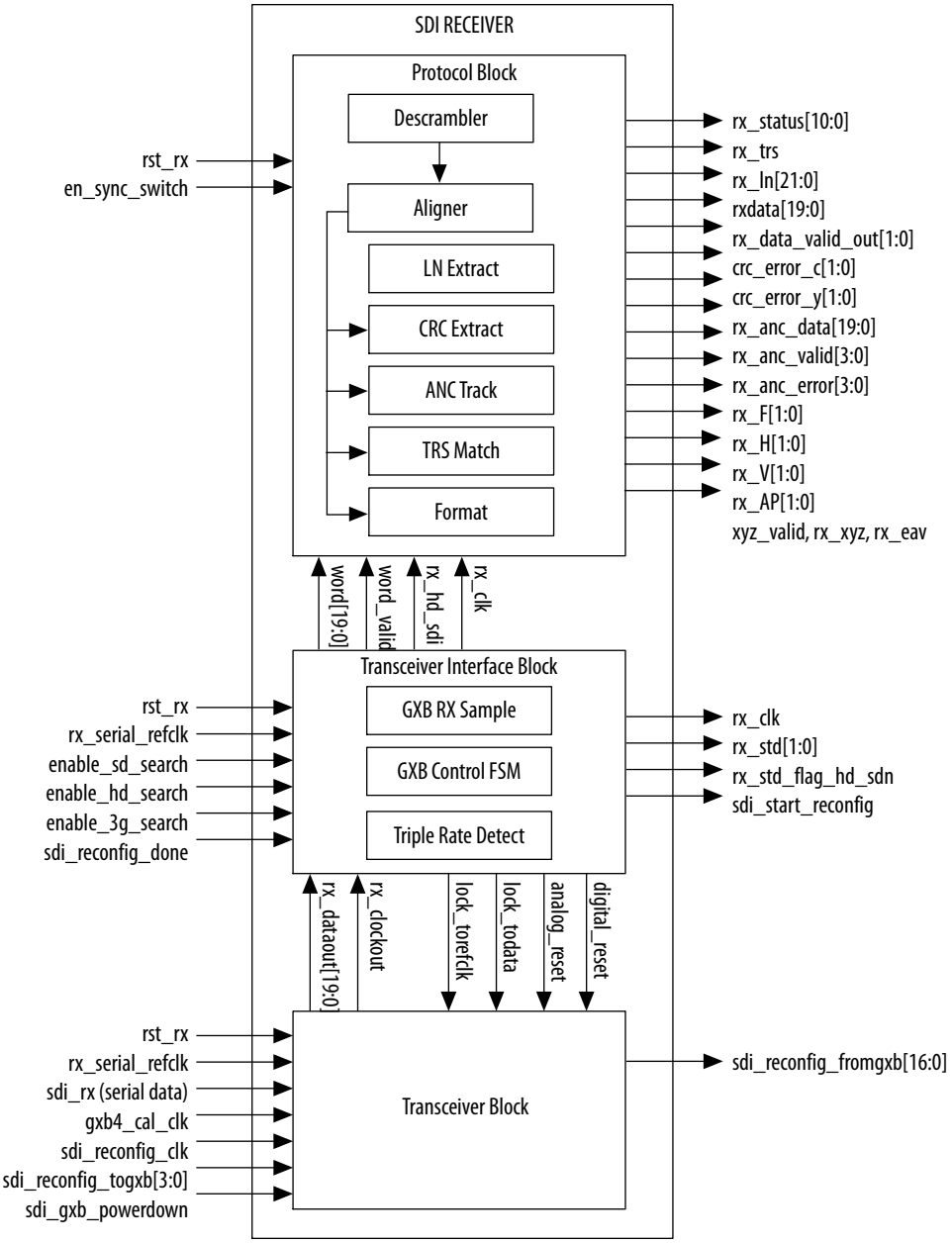
The receiver performs the following functions:

- NRZI decoding and descrambling
- Word alignment
- Video timing flags extraction
- RP168 switching compliance

- HD-SDI LN extraction
- HD-SDI CRC
- Accessing transceiver

Figure 6. SDI Receiver Block Diagram

The figure below shows the top-level block diagram for the SDI receiver.



The received data is NRZI decoded and descrambled and then presented as a word-aligned parallel output—20 bit for HD-SDI; 10 bit for SD-SDI (refer to Table 3–16 on page 3–41 for rxdata bus definition).

Table 8. Bit Allocation for rxdata for Supported Video Standards

The table below lists the bit allocation for `rxdata`.

rxdata	SD-SDI	HD-SDI	3G-SDI Level A	3G-SDI Level B
[19:10]	Unused	Y	Y	Cb, Y, Cr, Y multiplex (link A)
[9:0]	Cb, Y, Cr, Y multiplex	C	C	Cb, Y, Cr, Y multiplex (link B)

The receiver interface extracts and tracks the F, V, and H timing signals in the received data. Active picture and ancillary data words are also identified for your use.

For HD-SDI, the received CRC is checked for the luma and chroma channels. The LN is also extracted and provided as an output from the design.

NRZI Decoding and Descrambling

The descrambler module provides the channel decoding function that is common to both SDI and HD-SDI. It implements the NRZI decoding followed by the required descrambling. The algorithm indicated by SMPTE259M figure C.1 is iteratively applied to the receiver data, with the LSB processed first.

Word Alignment

The aligner word aligns the descrambled receiver data such that the bit order of the output data is the same as that of the original video data.

Table 9. EAV and SAV Sequences

The EAV and SAV sequences determine the correct word alignment. The table below lists the pattern for each standard.

Video Standard	EAV and SAV Sequences
SD-SDI	3FF 000 000
HD-SDI	3FF 3FF 000 000 000 000
3G-SDI Level A	3FF 3FF 000 000 000 000
3G-SDI Level A	3FF 3FF 3FF 3FF 000 000 000 000 000 000 000 000

The aligner matches the selected pattern in the descrambled receiver data. If the pattern is detected at any of the possible word alignments, then a flag is raised and the matched alignment is indicated. This process is applied continuously to the receiver data.

The second stage of the aligner determines the correct word alignment for the data. It looks for three consecutive TRSs with the same alignment, and then stores that alignment. If two consecutive TRSs are subsequently detected with a different alignment, then this new alignment is stored.

The final stage of the aligner applies a barrel shift function to the received data to generate the correctly aligned parallel word output. For this SDI MegaCore function, the barrel shifter allows the design to instantly switch from one alignment to another.

Video Timing Flags Extraction

The TRS match module extracts the F, V, and H video timing flags from the received data. You can use these flags for receiver format detection, or in the implementation of a flywheel function.

The TRS match module also identifies the line number and CRC words for HD-SDI.

RP168 Switching Compliance

To meet the RP168 requirements, the transceiver must be able to recover by the end of the switching line.

Table 10. Supported Video Switching Type

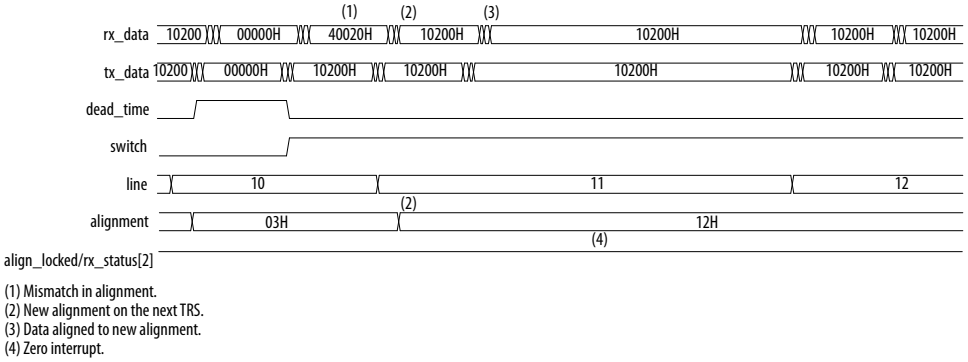
The table below lists the supported video switching type.

Standard/ Data Rate	Format	RP168 Support	Switching Source
Fixed	Switch (same format)	Yes	HD-1080i30 to HD-1080i30
Fixed	Switch	No	HD-1080 to HD-720
Switch	Fixed	No	HD-1080 to SD-525
Switch	Switch	No	HD-1080 to SD-525

The following figures show the behaviors of the aligner and format blocks during the RP168 switching.

Figure 7. Aligner Block Behavior

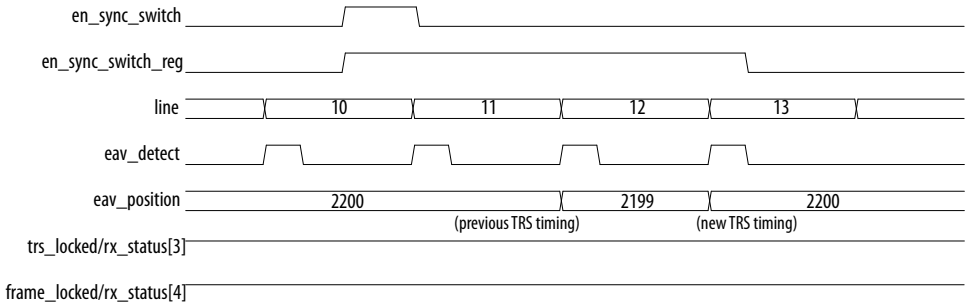
The aligner block immediately aligns to the next TRS timing based on the user input `en_sync_switch` signal.



(1) Mismatch in alignment.
(2) New alignment on the next TRS.
(3) Data aligned to new alignment.
(4) Zero interrupt.

The format block latches the user input `en_sync_switch` signal for three lines to realign to a new TRS alignment immediately. During switching, you see zero interrupt at downstream. The `trs_locked` and `frame_locked` signals never get deasserted during sync switch.

Figure 8. Format Block Behavior



HD-SDI LN Extraction

The HD-SDI LN extraction module extracts and formats the LN words defined by SMPTE292M section 5.4 from the HD-SDI chroma channel. The design provides the LN as an output.

HD-SDI CRC Checking

The CRC module checks the CRC defined by SMPTE292M section 5.5 for the HD-SDI luma and chroma channels.

Note: This module is common to the receiver and the transmitter.

The check is implemented by recalculating the CRCs for each received video line and then checking the results against the CRC data received. If the results differ, an error flag is asserted. There are separate error flags for the luma and chroma channels. The flag is held asserted until the next check is performed.

Accessing Transceiver

The Intel Quartus Prime Standard Edition software enables you to access the transceiver through the unencrypted ALTGX wrapper file. You can access the ALTGX wrapper files for Arria II GX, Arria V, Cyclone IV GX, and Stratix IV GX configurations.

You can use one of the following ways to access the ALTGX wrapper files:

- Edit the ALTGX wrapper file, using legal range provided in the respective device handbooks.
- Use analog control through the ALTGX_RECONFIG megafunction.

Note: Do not reinstantiate the customized ALTGX wrapper file using the IP catalog if you do not want to lose the default content of the wrapper file after regeneration.

Editing the ALTGX Wrapper File

To change the settings of the parameters, edit the legal ranges in the ALTGX wrapper file.

For example, to change the voltage output differential control setting from 4 to 7, change the following line in the wrapper file:

```
alt4gxb_component.vod_ctrl_setting = 4
```

to this line:

```
alt4gxb_component.vod_ctrl_setting = 7
```

Note: To know the exact legal ranges for a specific Intel device, refer to the respective device handbooks.

Using Analog Control

If you want the flexibility to access and control the ALTGX settings, use the ALTGX_RECONFIG megafunction to enable analog reconfiguration. Use the analog control to edit the default settings of the following transceiver parameters:

- Voltage output differential
- Pre-emphasis control pre-tap
- Pre-emphasis control 1st post-tap
- Pre-emphasis control 2nd post-tap
- Equalizer DC gain
- Equalizer DC control

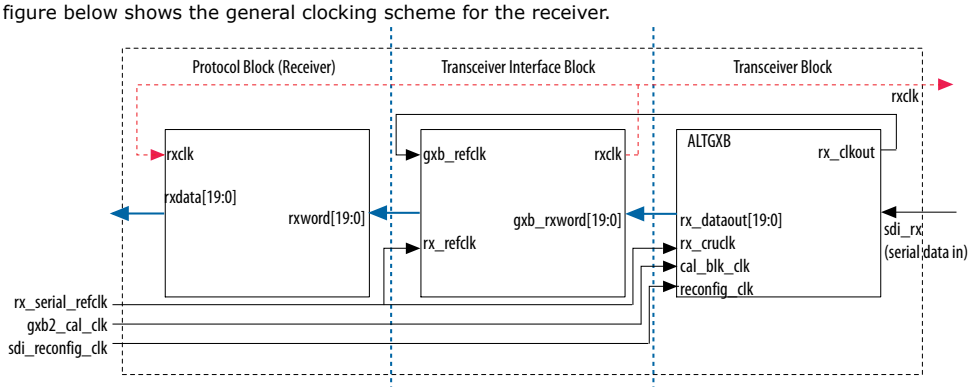
The ALTGX_RECONFIG megafunction connects with ALTGX using `reconfig_togxb[3:0]` and `reconfig_fromgxb[16:0]` ports for a single channel.

To enable the analog control and channel reconfiguration during run time, use the `reconfig_mode_sel` signal.

Transceiver Clock

Figure 9. Receiver Clocking Scheme

The figure below shows the general clocking scheme for the receiver.



3.3. Transceiver

The transceiver deserializes the high-speed serial input.

For HD-SDI, the CDR function performs the deserialization and locks the receiver PLL to the receiver data.

For SD-SDI, the transceiver provides a fixed frequency oversample of the serial data with the receiver PLL constantly locked to a reference clock, which allows the transceiver to support the 270-Mbps data rate.

The transceiver can process either SD-SDI or HD-SDI data. The data rate can be automatically detected so that the interface can handle both SD-SDI and HD-SDI without the need for device reconfiguration.

Features	Supported devices
Two transmitter PLLs per quad. Each quad allows two independent transmitter rates. Receivers in a quad share a common training clock, but have independent receiver PLLs. Because the same training clock is used for SD-SDI and HD-SDI, receivers can accommodate the different standards within a single quad.	Arria II GX, Arria V, Stratix IV GX, and Stratix V
Additional serial reference clock port. This additional clock port allows you to have two different clock rates for different data rates using a single transceiver block, with the ability to switch between the desired clock rates (for example, 148.5 MHz and 148.35 MHz).	Arria II GX (including Arria II GZ) and Stratix IV GX
Eight regular transceiver channels from the upper and lower quads. There are four MPLLs and two GPLLs that you can use to clock the transceiver channels. Each receiver in EP4CGX50 and EP4CGX75 devices has a clock divider, which allows one MPLL to drive all the receiver channels. The receiver in EP4CGX110 and EP4CGX150 devices does not have a clock divider, which limits each MPLL to drive only one receiver channel to accommodate the different standards within a single quad. You must supply two receive reference clocks (for example, 148.5 MHz and 148.35 MHz) to the SDI receiver. Implement the PPM detection function in the user logic to detect the ppm difference between the receive reference clock and the recovered clock. Based on the difference detected, you must switch between the two receive reference clocks by toggling the `rx_serial_refclk_clkswitch` signal.	Cyclone IV GX devices—EP4CGX30 (F484), EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150

Related Information

V-Series Transceiver PHY IP Core User Guide Notes

3.3.1. Transmitter Clocks

The transmitter requires two clocks: a parallel video clock (`tx_pclk`) and a transmitter reference clock (`tx_serial_refclk`).

The parallel video clock samples and processes the following parallel video input:

- SD-SDI—27 MHz
- HD-SDI—74.25 or 74.175 MHz
- 3G-SDI—148.5 or 148.35 MHz

The transceiver uses the transmitter reference clock to generate the high-speed serial output. The transceiver is configured for 20-bit operation, so the reference clock is 1/20th of the serial data rate.

Note: For SD-SDI, because of the oversampling implementation, the serial data rate is five times the SDI bit rate (for example, 1,350 Mbps); for the triple-standard SDI, the oversampling rate is 11.

For SD-SDI operation, the transmitter reference clock can be derived from `pclk` by using one of the transceiver PLLs. The PLL can multiply the 27-MHz pclk signal by 5/2.

For all other standards, use an external multiplexer to select between the alternative reference clocks.

Table 11. Transmitter Clock Frequency

This table lists the frequencies of the transmitter clock, `tx_serial_refclk`, for Arria II GX, Arria V, Cyclone IV GX, Cyclone V, Stratix IV, and Stratix V devices.

Video Standard	Clock Frequency (MHz)
SD-SDI	67.5
HD-SDI (including dual link)	74.175 or 74.25
HD-SDI with two times oversample	148.35 or 148.5
Dual standard	67.5, 74.175 or 74.25
Tripe standare	148.35 or 148.5
3G-SDI	148.35 or 148.5

Note: You must multiplex the `tx_serial_refclk` signal externally for all the video standards, except for SD-SDI,. If you enable the additional input reference clock port for the serial reference clock, the external multiplier is no longer required.

The following figures show the transmitter clocks for Arria II GX, Arria V, Cyclone IV GX, Cyclone V, Stratix IV, and Stratix V devices.

Figure 10. Transmitter Clock for SD-SDI

SD-SDI

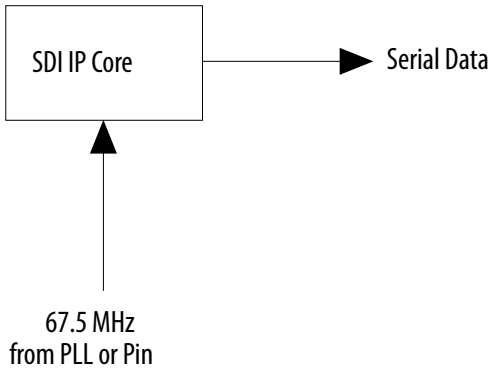


Figure 11. Transmitter Clock for HD-SDI

The frequency supported for this clock can be 74.175 or 74.25 MHz, to support 1.4835 or 1.485 Gbps HD-SDI respectively.

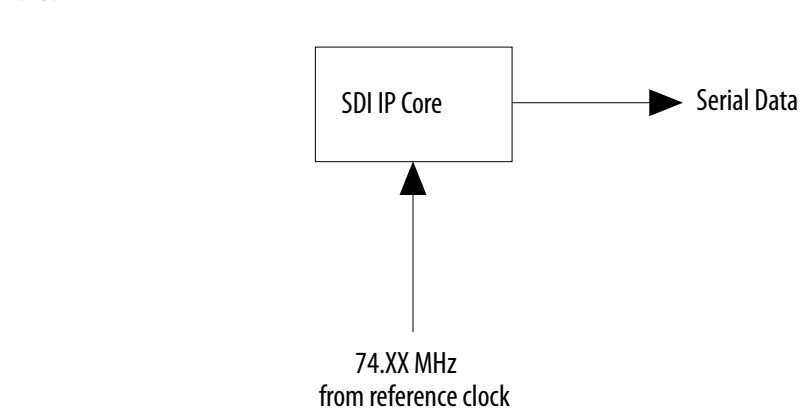


Figure 12. Transmitter Clock for Dual Standard

The multiplexer must be in the device.

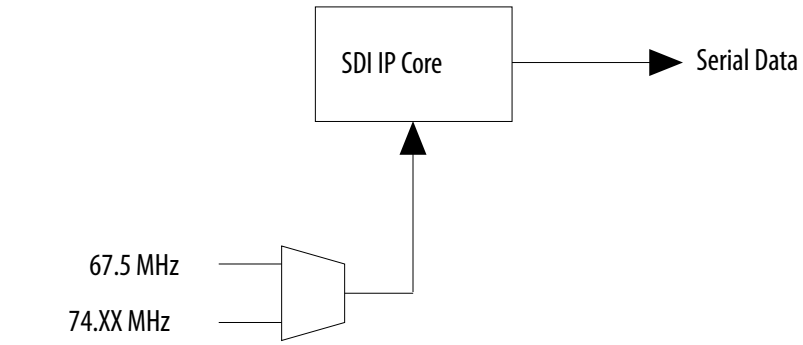


Figure 13. Transmitter Clock for 3G-SDI and Triple Standard

The frequency supported for this clock can be either 148.35 or 148.5 MHz, to support 2.967 or 2.970 Gbps HD-SDI respectively

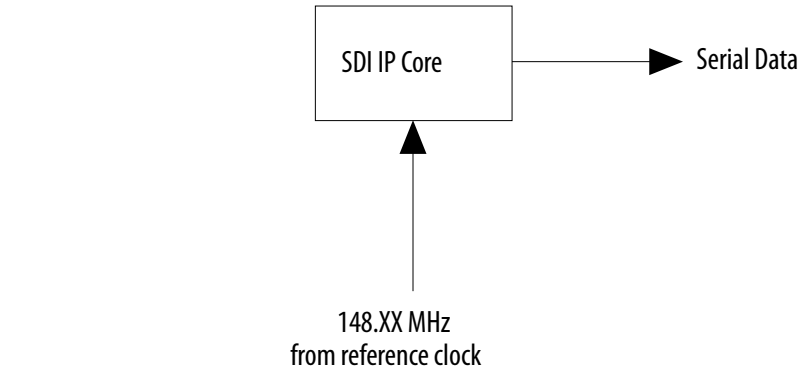
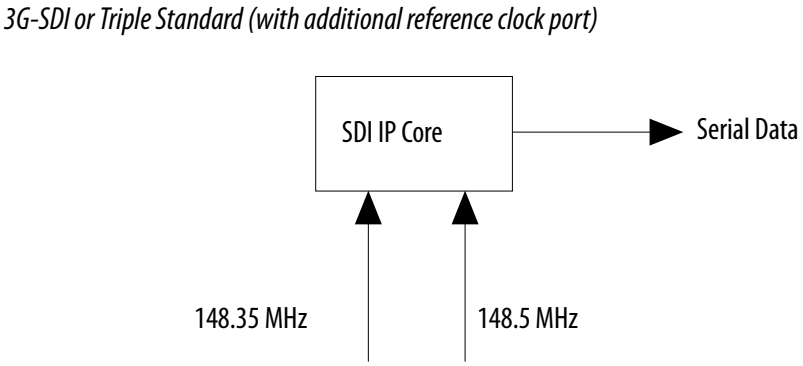


Figure 14. Transmitter Clock for 3G-SDI and Triple Standard with Additional Reference Clock Port

You can source both 148.5 MHz and 148.35 MHz clocks together if the additional clock port is enabled.



Related Information

Altera Transceiver PHY IP Core User Guide
More information about the Altera Native PHY IP Core.

3.3.2. Receiver Clocks

The transceiver requires a receiver reference clock, `rx_serial_refclk`. This clock trains the receiver PLL in the transceiver.

- For SD-SDI operation, the clock must be nominally 1/4th of the serial data rate (for example, 67.5 MHz). The clock does not have to be frequency locked to the data.
- For HD-SDI operation, the clock must be nominally 1/20th of the serial data rate. The clock does not have to be frequency locked to the data, because the design only uses it for the training of the receiver PLL.
- For dual or triple standard operation, the receiver reference clock must be 148.5 MHz. In this mode, the transceiver oversamples the SD-SDI signals by a factor of 11.

All receiver interfaces share a common receiver reference clock.

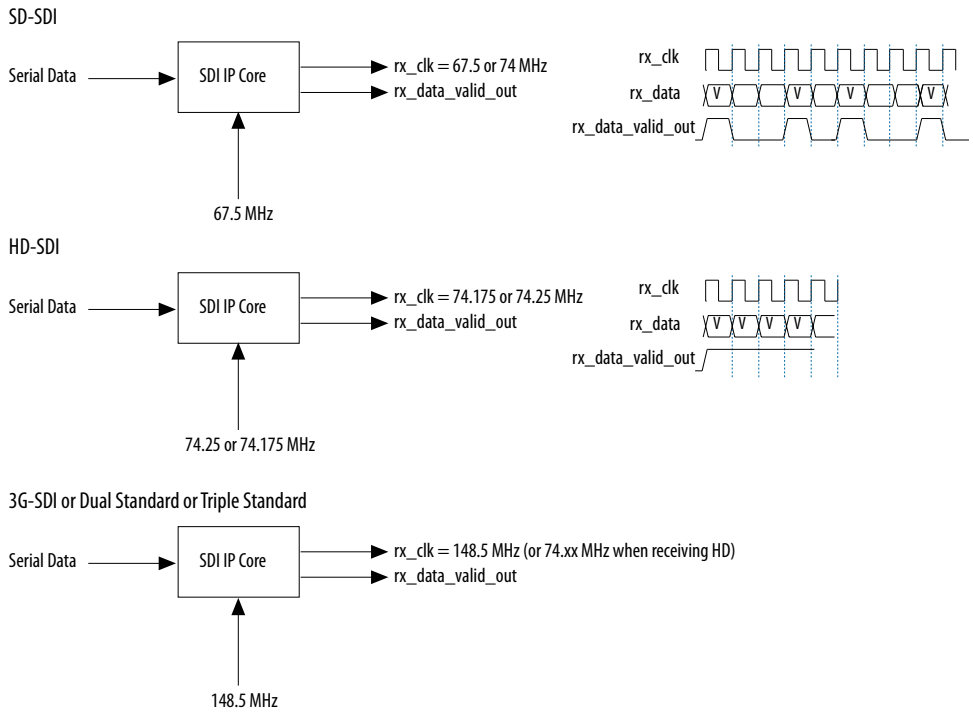
Table 12. Receiver Clock Frequency

This table shows the frequencies of the receiver clock, `rx_serial_refclk`.

Video Standard	Clock Frequency (MHz)
SD-SDI	67.5
HD-SDI	74.175 or 74.25
Dual or triple standard	148.35 or 148.5(6)
3G-SDI	148.35 or 148.5

(6) For correct SD-SDI operation, you must use only 148.5 MHz

Figure 15. Receiver Clocks for Different Standards



3.3.3. Transmitter Transceiver Interface

Intel provides a transceiver interface, which interfaces the transceiver to the SDI function.

The transmitter transceiver interface implements the following functions:

- Retiming from the parallel video clock domain to the transceiver transmitter clock domain.
- Optional two-times oversampling for HD-SDI
- Transmitter oversampling for SD-SDI

Table 13. Transmitter Transceiver Interface Functions

Functions	Description
Transmitter retiming	The parallel data input, `txdata`, to the transceiver must be synchronous and phase-aligned to the transceiver clock input, `tx_coreclk`. SD-SDI (and optionally HD-SDI) requires a retiming function, because of the oversampling logic. The transmitter uses a small 16×20 FIFO buffer for the retiming.

continued...

Functions	Description
	• For HD-SDI, the FIFO buffer realigns the parallel video input to the transceiver clock, `tx_coreclk`. It is written on every `tx_pclk`, and read on every `tx_coreclk`. • For SD-SDI, the FIFO buffer also provides the rate conversion required by the transmitter oversampling logic. It is written on every other `tx_pclk`, using the SD-SDI data width conversion logic. It is read on every 5th or 11th `tx_coreclk`. This operation ensures that the transmitter oversampling logic is provided with a word of parallel video data on every 5th or 11th clock.
HD-SDI two-times oversampling	The two-time oversampling mode performs two-times oversampling and runs the transceiver at double rate, to give a better output jitter performance. This mode requires a higher rate reference clock.
SD-SDI transmitter oversampling	SD-SDI requires a 270-Mbps serial data rate, which is achieved by transmitting a 1,350 Mbps signal with each bit repeated five times. This process ensures that the transceiver runs at a supported frequency. In triple standard mode, bit are transmitted at 2,970 Mbps with each bit repeated 11 times.

3.3.4. Receiver Transceiver Interface

Intel provides a transceiver interface, which interfaces the transceiver to the SDI function.

The receiver transceiver interface implements the following functions:

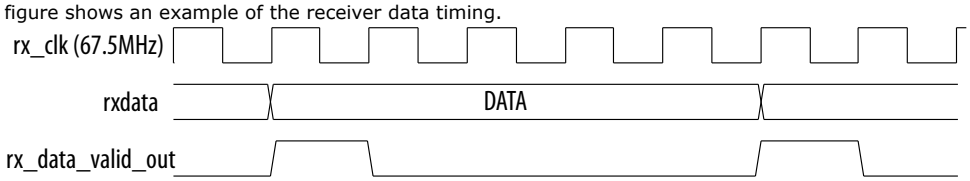
- Receiver oversampling for SD-SDI
- Transceiver controller

SD-SDI Receiver Oversampling

Arria II GX, Arria V, Stratix IV, and Stratix V transceivers do not support CDR for data rates less than 600 Mbps. The receiver uses fixed frequency oversampling for the reception of 270-Mbps SD-SDI. The transceiver samples the serial data at 1,350 or 2,970 Mbps and the SD-SDI receiver oversampling logic extracts the original 270 Mbps data.

Figure 16. Receiver Data Timing

This figure shows an example of the receiver data timing.



Transceiver Controller

To achieve the desired receiver functionality for the SDI, the transceiver controller controls the transceiver.

When the interface receives SD-SDI, the transceiver receiver PLL locks to the receiver reference clock.

When the interface receives HD-SDI, the transceiver receiver PLL is first trained by locking to the receiver reference clock. When the PLL is locked, it can then track the actual receiver data rate. If a period of time passes without a valid SDI signal, the PLL is retrained with the reference clock and the process is repeated.

First, the transceiver controller makes a coarse rate detection of the incoming data stream. Through transceiver dynamic reconfiguration, the transceiver is then reprogrammed to the correct rate for the standard detected. After the reprogramming, the transceiver attempts to lock to the incoming stream. If no valid data is seen in 0.1 s, the transceiver resets the receiver path and performs rate detection again.

At the start of the rate detection process, the level of the three `enable_xx` signals is sampled. The level of these signals and the knowledge of the currently programmed state of the transceiver determines if the transceiver requires programming. This process ensures that the transceiver is reprogrammed only when necessary.

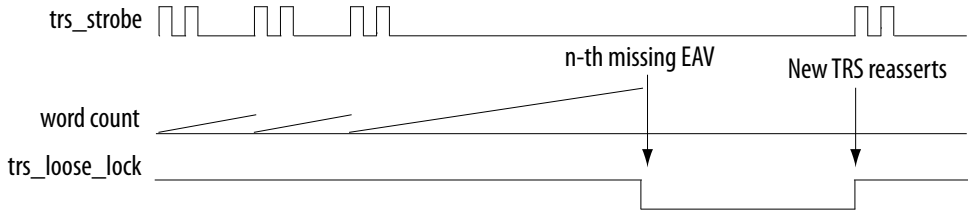
3.4. Locking to the Incoming SDI Stream

The transceiver control state machine uses the presence (or absence) of TRSs on the stream to determine if SDI is being correctly received.

A single, valid TRS indicates to the control state machine that the receiver is acquiring some valid SDI samples. The control state machine only deasserts this flag when it does not detect any EAV sequences within the number of consecutive lines you specified. At this point, the controller state machine resets and performs the relock algorithm.

Figure 17. Locking Algorithm

This figure shows how the controller state machine resets and performs the relock algorithm.



Because the aligner realigns to a new alignment if two consecutive TRSs with the same alignment are detected, this scheme allows for an SDI source switch and an alignment change without affecting the transceiver reset state machine.

The SDI MegaCore function also monitors the incoming EAV and SAV signals to ensure their spacing is consistent over a number of lines. The MegaCore function monitors by incrementing a counter on each incoming SDI word and storing the count values at which an EAV or SAV is detected. If the EAV and SAV spacing is consistent over 6 video lines, the MegaCore function indicates `trs_locked` on the `rx_status[3]` output.

An enhancement in the current SDI MegaCore function allows a number of missing EAV or SAV that you specify to be tolerated without deasserting the `trs_locked` signal.

For example, when you specify the **Tolerance to consecutive missed EAV/SAV** parameter to **2**, one or two consecutive missing EAVs set a "missed" flag but do not cause the `trs_locked` signal to deassert. A good EAV in the correct position resets the "missed" flag.

The following figures show examples of the operation missing or misplaces TRS tolerance.

Figure 18. Single Missing EAV

This figure shows how a single missing EAV does not cause the `trs_locked` signal to deassert.

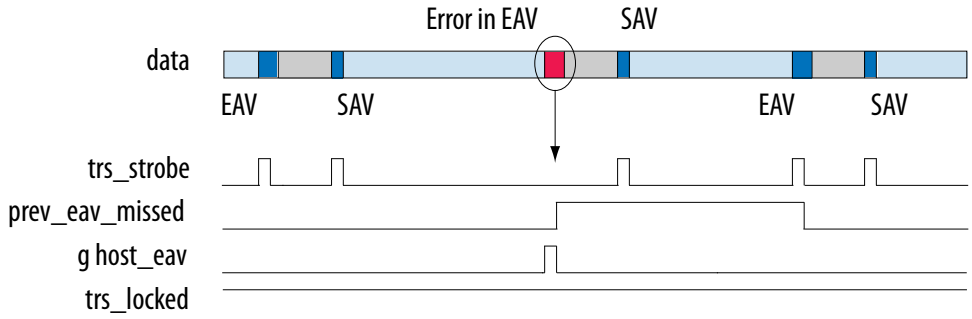


Figure 19. Two Consecutive Missing EAVs

This figure shows how two consecutive missing EAVs do not cause the `trs_locked` signal to deassert.

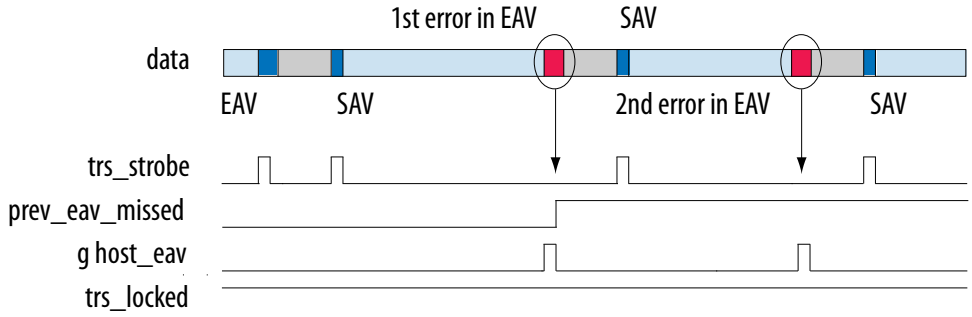
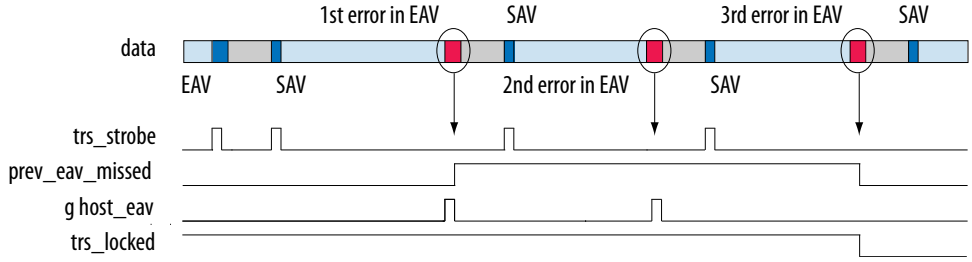


Figure 20. Three Consecutive Missing EAVs

This figure shows how three consecutive missing EAVs cause the `trs_locked` signal to deassert.



The `frame_locked` signal detects TRS EAV, inspects the transition of field (F) and vertical (V) synchronizations, and then counts the line number. The inspecting transitions on the F and V synchronizations provide the frame timing. The line count value is stored if there is a rising or falling edge on the F and V synchronizations through the frame. The stored count values are compared over multiple frames to make sure they are stable, before the `frame_locked` signal is asserted.

The `frame_locked` signal deasserts when there are bad F or V synchronizations, or when there is a rising edge from frame to frame. The `frame_locked` signal also deasserts when the `trs_locked` signal deasserts.

When the `frame_locked` signal is zero, the frame is invalid, and the receiver is not considered to receive reliable video data.

3.5. SDI Receiving Video Format Specification

The table below lists the 8-bit receiving video format specification .

Video Standard	Total Active Lines	Word per Total Line	Rate	rx_video_format [7:5]	rx_video_format [4]	rx_video_format [3:0]
					Progressive/ Interlace	Frame Rate
SD	720	—	—	0	0	8
720p60		1650	60	2	1	7
720p59.94			59.94			6
720p50		1980	50			5
720p30		3300	30			4
720p29.97			29.97			3
720p25		3960	25			2
720p24		4125	24			1
720p23.97			23.97			0
1035i30	1035	2200	30	3	0	4
1035i29.97			29.97	3	0	3
1080i25	1080	2376	25	4	0	2
1080i60		2200	60	1		7
1080i59.94			59.94			6
1080i50		2640	50			5
1080i24		2750	24			1
1080i23.97			23.97			0
1080p60	1080	2200	60	1	1	7
1080p59.94			59.94			6
1080p50		2640	50			5
1080p30	1080	2200	30	1	1	4
1080p29.97			29.97			3
1080p25		2640	25			2
1080p24		2750	24			1
1080p23.97			23.97			0

4. SDI IP Core Signals

4.1. SDI Clock Signals

Table 14. Receiver Clock Signals

Signal	Direction	Description
gxb4_cal_clk	Input	Calibration clock for Arria II GX, Arria V, Cyclone IV GX, and Stratix IV transceivers only.
rx_sd_oversample_clk_in	Input	67.5-MHz oversample clock input. SD-SDI only.
rx_serial_refclk	Input	Transceiver training clock for HD-SDI, dual standard and triple standard. *Note:* You must tie the tx_serial_refclk and rx_serial_refclk signals together if you generate an SDI duplex using the Stratix V or Arria V devices.
rx_serial_refclk1	Input	Secondary transceiver training clock. Clock frequency of 74.175 MHz for HD-SDI, or clock frequency of 148.35 MHz for 3G-SDI, dual standard and triple standard. Available only when you use a Cyclone IV GX device.
rx_coreclk	Input	Receiver controller clock input. For Cyclone IV GX devices only. The frequency of this clock must be the same as rx_serial_refclk. Because of hardware constraint, the transceiver PLL and core logic cannot share the same clock input pin if they use transceiver PLL6 and PLL7.
refclk_rate	Input	This signal is related to the rx_video_format signal. Detects the received video standard. Set input to 0 for a 148.35-MHz receiver serial reference clock. Set input to 1 for 148.5-MHz RX serial reference clock. *Note:* For Cyclone IV GX devices, set the refclk_rate according to the rx_coreclk frequency.
gxb_tx_clkout	Output	Transmitter clock out of transceiver. This clock is the output of the voltage-controlled oscillator (VCO) and is used as a parallel clock for the transmitter. It connects internally to the tx_clkout signal of the ALTGX or ALT2GXB megafunction.
rx_clk	Output	Transceiver CDR clock.
rx_sd_oversample_clk_out	Output	67.5-MHz oversample clock output for cascading MegaCore functions. SD-SDI only.
rx_video_format	Output	This signal is related to the refclk_rate signal. Indicates the format for the received video. The rx_video_format value is valid after the frame locked signal is asserted.

Table 15. Transmitter Clock Signals

Signal	Direction	Description
tx_pclk	Input	Transmitter parallel clock input.

continued...

Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

Signal	Direction	Description
		• SD-SDI = 27 MHz • HD-SDI = 74 MHz • 3G-SDI = 148.5 MHz
tx_serial_refclk	Input	Transceiver reference clock input; with low jitter. *Note:* You must tie the tx_serial_refclk and rx_serial_refclk signals together if you generate an SDI duplex using the Stratix V or Arria V devices.
tx_serial_refclk1	Input	Optional port for transceiver reference clock input; with low jitter. Similar to tx_serial_refclk. *Note:* Only available for Arria II, and Stratix IV GX devices.

Table 16. Transceiver PHY Management Signals

These signals are only available for Arria V and Stratix V devices.

Signal	Direction	Description
phy_mgmt_clk	Input	Avalon-MM clock input for the transceiver PHY management interface. Use the same clock for the PHY management interface and transceiver reconfiguration. The frequency range is 100-125 MHz to meet the specification of the transceiver reconfiguration clock.
phy_mgmt_clk_reset	Input	Reset signal for the transceiver PHY management interface. This signal is active high and level sensitive. This signal can be tied to the same reset port as tx_rst or rx_rst signal in simplex mode. In duplex mode, this reset signal acts as a global reset for both the transmitter and receiver. If you require a different reset for the transmitter and receiver, separate this signal from the tx_rst and rx_rst signal.

Table 17. Soft Transceiver Signals

rx_sd_refclk_337	Input	Soft transceiver 337.5-MHz sampling clock.
rx_sd_refclk_337_90deg	Input	Soft transceiver 337.5-MHz sampling clock with 90° phase shift.
rx_sd_refclk_135	Input	Soft transceiver 135-MHz parallel clock for receiver.
rx_sd_refclk_270	Input	Soft transceiver 270-MHz parallel clock for transmitter.

4.2. SDI Interface Signals

Signal	Width	Direction	Description
enable_crc	[(N–1):0]	Input	Enables CRC insertion for HD-SDI and 3G-SDI.
enable_sd_search	[1:0]	Input	Enables search for SD-SDI signal in dual or triple standard mode.
enable_hd_search	[1:0]	Input	Enables search for HD-SDI signal in dual or triple standard mode.
enable_3g_search	[1:0]	Input	Enables search for 3G-SDI signal in triple standard mode.
enable_ln	[(N– 1):0]	Input	Enables line number (LN) insertion for HD-SDI and 3G-SDI.
en_sync_switch	[1:0]	Input	Enables aligner and format blocks to realign immediately so that the downstream is completely non-disruptive.

continued...

Signal	Width	Direction	Description
rst_rx	[1:0]	Input	Reset signal, which holds the receiver in reset. For Cyclone IV GX devices, this signal must be synchronous to the rx_coreclock clock domain for the receiver. Issues a reset to the SDI IP core after power-up to ensure reliable operation. For HD-SDI dual link receiver, assert this signal when both link A and link B are ready for the first time.
rst_tx	[1:0]	Input	Reset signal, which holds the transmitter in reset. The reset synchronization for the transmitter is handled within the SDI IP core. *Note:* The video mode (tx_std) and clocks must be set up and stable before device bring-up or core reset. Issues a reset to the SDI IP core after power-up to ensure reliable operation.
rx_serial_refclk_clkswitch	[1:0]	Input	Reference clock switching. Available only when you use a Cyclone IV GX device. Toggle between rx_serial_refclk and rx_serial_refclk1 at every positive edge triggered.
rx_protocol_clk	[(N–1):0]	Input	External clock for protocol data.
rx_protocol_hd_sdn	[(N–1):0]	Input	Selection of HD-SDI or SD-SDI processing for dual or triple standard protocol block. This signal only appears on dual or triple standard protocol blocks and indicates 3G-SDI(1), HD-SDI(1) or SD-SDI(0) data on the rx_protocol_in signal. You must connect this signal to the rx_std_flag_hd_sdn output of the transceiver block in a split protocol/transceiver design.
rx_protocol_in	[(20N–1):0]	Input	External data input for protocol only mode.
rx_protocol_locked	[(N–1):0]	Input	Input to transceiver control logic. When active, this signal indicates to the transceiver control logic that the protocol blocks are locked, to stop the transceiver search algorithm at the current rate.
rx_protocol_rst	[(N–1):0]	Input	Reset for the protocol block. This signal resets the protocol blocks. You can connect this signal to the rx_status[1] pin (sdi_reset) in a split transceiver/protocol design.
rx_protocol_valid	[(N–1):0]	Input	External data valid in for protocol only mode.
rx_protocol_rate	[1:0]	Input	Input to the protocol block. This signal indicates the received video standard to the protocol block. However, this signal does not distinguish between 3G-SDI Level A and 3G-SDI Level B streams. The aligner block in the protocol block distinguishes the 3G-SDI Level A and 3G-SDI Level B streams. You must connect this signal to the rx_std port of the transceiver block in a split transceiver/protocol design.
rx_xcvr_trs_lock	[(N–1):0]	Input	Input to transceiver control logic. You must connect this signal to the rx_status[3] pin (trs_locked) of the protocol only receiver block.
sdi_rx	[(N–1):0]	Input	Serial input.
txdata	[(20N–1):0]	Input	User-supplied transmitter parallel data valid. SD-SDI uses 9:0; HD-SDI uses 20N – 1:0.

continued...

Signal	Width	Direction	Description
			• SD-SDI = bits 19:10 unused; bits 9:0 Cb, Y, Cr, Y multiplex • HD-SDI = bits 19:10 Y; bits 9:0 C • Dual Link = bits 39:30 Y link B; bits 29:20 C link B; bits 19:10 Y link A, bits 9:0 C link A • 3G-SDI Level A = bits 19:10 Y; bits 9:0 C • 3G-SDI Level B = bits 19:10 Cb, Y, Cr, Y multiplex (link A); bits 9:0 Cb, Y, Cr, Y multiplex (link B)
tx_ln	[21:0]	Input	Transmitter line number. • HD-SDI = bits 21:11 11'd0; bits 10:0 LN • Dual Link = bits 21:11 LN link B; bits 10:0 LN link A • 3G-SDI Level A = bits 21:11 11'd0; bits 10:0 LN • 3G-SDI Level B = bits 21:11 LN link A; bits 10:0 LN link B
tx_trs	[(N–1):0]	Input	Transmitter TRS input. For use in HD-SDI LN and CRC insertion. Assert on first word of both EAV and SAV TRSs.
tx_std	[1:0]		Transmitter standard. • SD-SDI = 00 • HD-SDI = 01 • 3G-SDI Level A = 11 • 3G-SDI Level B = 10 *Note:* This signal must be set up and stable prior to device bring-up or core reset.
trs_loose_lock	[(N–1):0]	Output	TRS locking signal for protocol only receiver mode. You can connect this signal to the rx_protocol_locked pin of the transceiver only receiver block.
crc_error_y	[1:0]	Output	CRC error on luma channel. • HD-SDI: bit 1 unused; bit 0 crc_error_y • Dual link: bit 1 link B crc_error_y; bit 0 link A crc_error_y • 3G-SDI Level A: bit 1 unused; bit 0 crc_error_y • 3G-SDI Level B: bit 1 link A crc_error_y; bit 0 link B crc_error_y
crc_error_c	[1:0]	Output	CRC error on chroma channel. • HD-SDI: bit 1 unused; bit 0 crc_error_c • Dual link: bit 1 link B crc_error_c; bit 0 link A crc_error_c • 3G-SDI Level A: bit 1 unused; bit 0 crc_error_c • 3G-SDI Level B: bit 1 link A crc_error_c; bit 0 link B crc_error_c
rx_AP	[1:0]	Output	This is an active picture interval timing signal. The receiver asserts this signal when the active picture interval is active. • SD-SDI/HD-SDI: bit 1 unused; bit 0 rx_ap • Dual link: bit 1 link B unused; bit 0 link A rx_ap • 3G-SDI Level A: bit 1 unused; bit 0 rx_ap • 3G-SDI Level B: bit 1 link A rx_ap; bit 0 link B rx_ap
rxdata	[(20N–1):0]	Output	Receiver parallel data. SD-SDI uses 9:0; HD-SDI uses 20N–1:0.

continued...

Signal	Width	Direction	Description
			• SD-SDI bits 19:10 unused; bits 9:0 Cb, Y, Cr, Y multiplex • HD-SDI bits 19:10 Y; bits 9:0 C • Dual link: bits 39:30 Y link B; bits 29:20 C link B; bits 19:10 Y link A, bits 9:0 C link A • 3G-SDI Level A: bits 19:10 Y; bits 9:0 C • 3G-SDI Level B: bits 19:10 Cb, Y, Cr, Y multiplex (link A); bits 9:0 Cb, Y, Cr, Y multiplex (link B)
rx_data_valid_out	[1:0]	Output	Data valid from the oversampling logic. Asserted to indicate current data on rxdata is valid. Bit 0 of this bus indicates valid data on rxdata. When receiving SMPTE 425M-B signals in 3G-SDI or triple standard, bit 1 indicates that data on rxdata is from virtual link A; bit 0 indicates the data is from virtual link B.
rx_F	[1:0]	Output	This is a field bit timing signal. This signal indicates which video field is currently active. For interlaced frame, 0 means first field (F0) while 1 means second field (F1). For progressive frame, the value is always 0. • SD-SDI/HD-SDI: bit 1 unused; bit 0 rx_f • Dual link: bit 1 unused; bit 0 rx_f • 3G-SDI Level A: bit 1:0 unused • 3G-SDI Level B: bit 1:0 unused
rx_H	[1:0]	Output	This is a horizontal blanking interval timing signal. The receiver asserts this signal when the horizontal blanking interval is active. • SD-SDI/HD-SDI: bit 1 unused; bit 0 rx_h • Dual link: bit 1 unused; bit 0 rx_h • 3G-SDI Level A: bit 1 unused; bit 0 rx_h • 3G-SDI Level B: bit 1 link A rx_h; bit 0 link B rx_h
rx_ln	[21:0]	Output	Receiver line number. • HD-SDI = bits 21:11 unused; bits 10:0 LN • Dual Link = bits 21:11 unused; bits 10:0 LN • 3G-SDI Level A = bits 21:11 unused; bits 10:0 LN • 3G-SDI Level B = bits 21:11 LN link A; bits 10:0 LN link B
rx_std_flag_hd_sdn	1	Output	Indicates received standard for dual or triple standard only. HD-SDI = 1; SD-SDI = 0.
rx_V	[1:0]	Output	This is a vertical blanking interval timing signal. The receiver asserts this signal when the vertical blanking interval is active. • SD-SDI/HD-SDI: bit 1 unused; bit 0 rx_v • Dual link: bit 1 unused; bit 0 rx_v • 3G-SDI Level A: bit 1 unused; bit 0 rx_v • 3G-SDI Level B: bit 1 link A rx_v; bit 0 link B rx_v
rx_xyz	1	Output	Receiver output that indicates current word is XYZ word.
xyz_valid	1	Output	Receiver output that indicates current TRS format is legal (XYZ word is correct).
rx_eav	1	Output	Receiver output that indicates current TRS is EAV.

continued...

Signal	Width	Direction	Description
rx_trs	1	Output	Receiver output that indicates current word is TRS. This signal is asserted at the first word of 3FF 000 000 TRS.
sdi_tx	[(N–1):0]	Output	Serial output.
tx_protocol_out	[(20N–1):0]	Output	Data out (protocol only mode).

The following figures illustrate the input and output interface signals for SDI triple standard instances.

Figure 21. Interface Signals for SDI Triple Standard Receiver

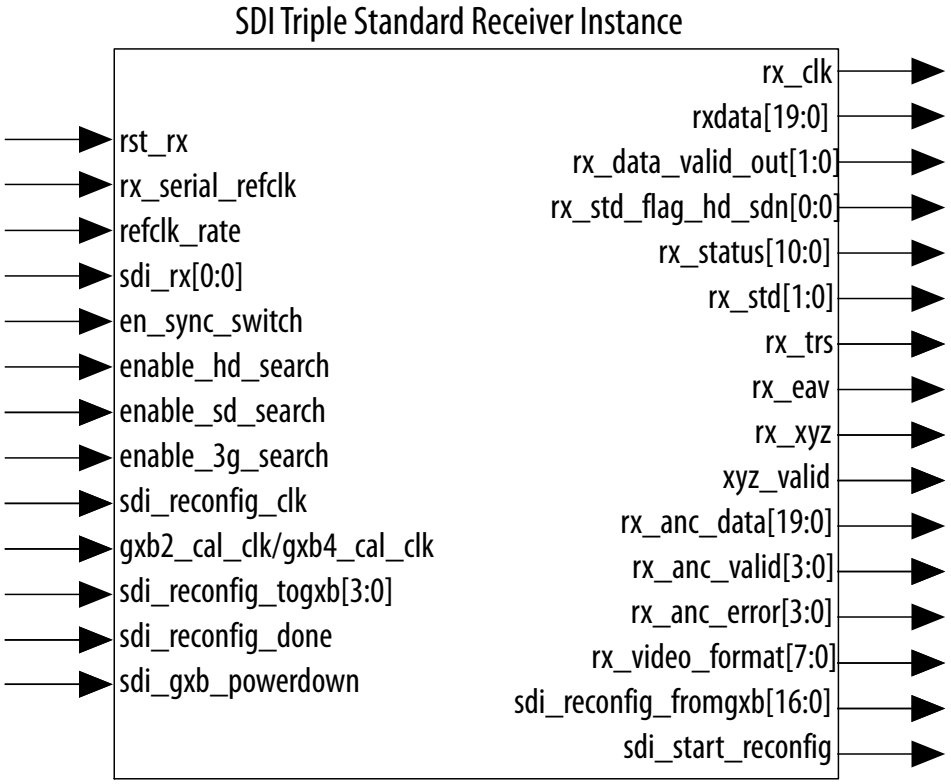


Figure 22. Interface Signals for SDI Triple Standard Transmitter

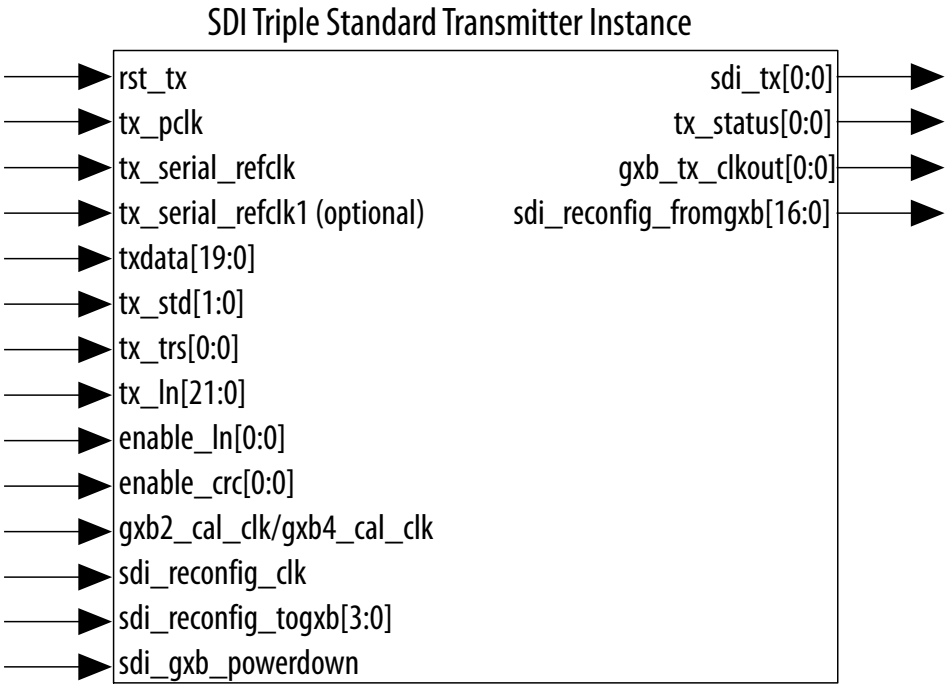
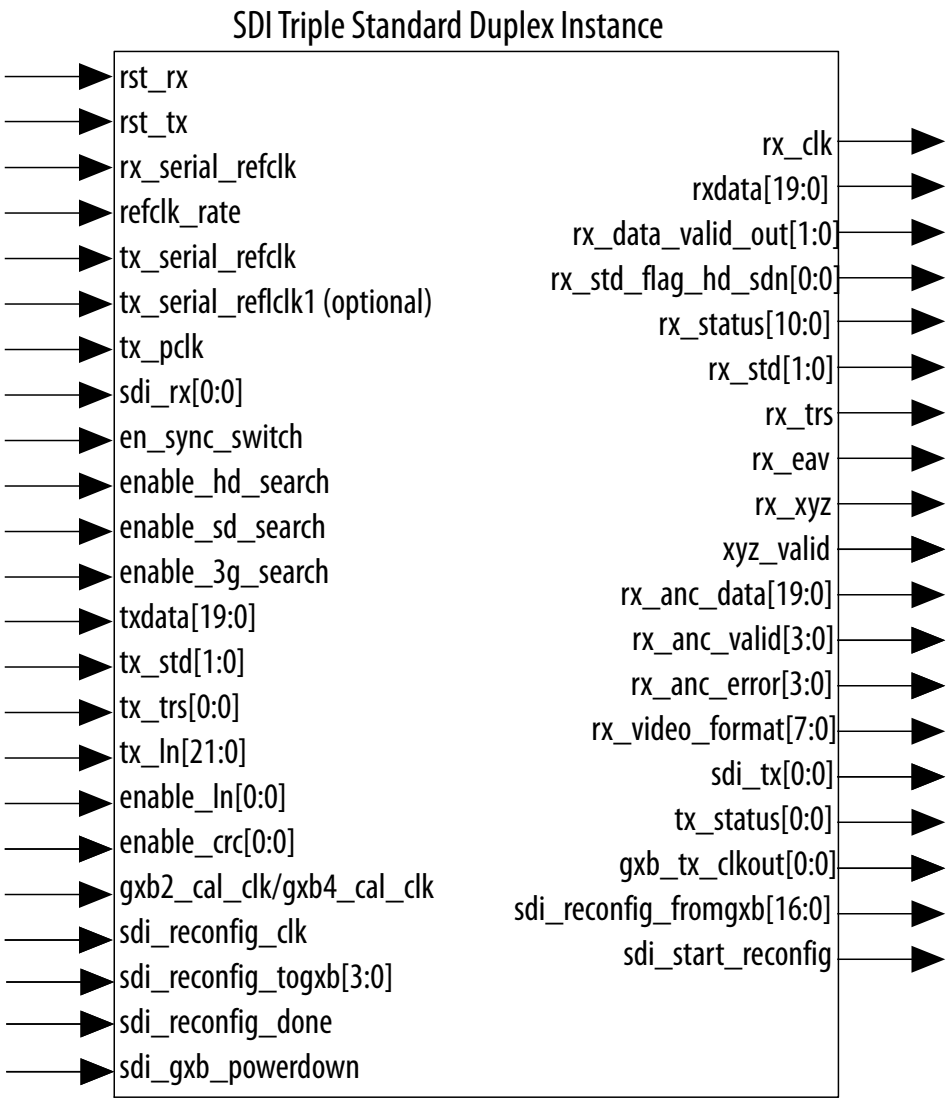


Figure 23. Interface Signals for SDI Triple Standard Duplex Instance



The following figures show the behavior of certain SDI interface signals.

Figure 24. Power-up Reset for the Receiver

Figure 25. Power-up Reset for the Transmitter

`rst_tx_sync` is an internally synchronized reset for the transmit circuits.

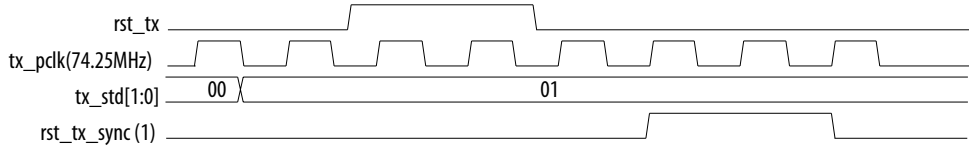


Figure 26. Behavior of tx_std, tx_trs, and tx_ln Signals—425MA

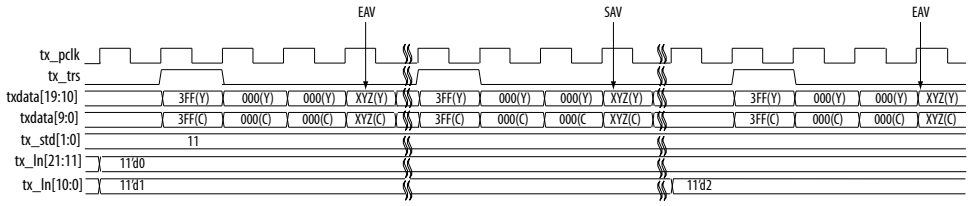


Figure 27. Behavior of tx_std, tx_trs, and tx_ln Signals—425MB

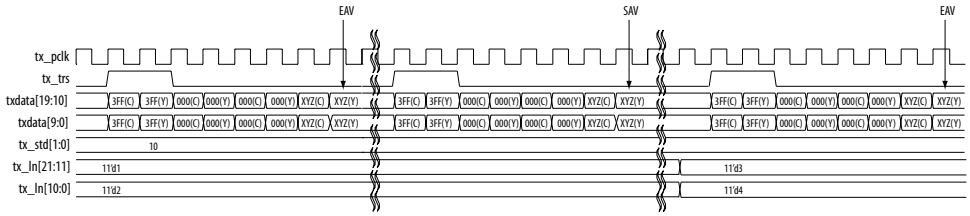
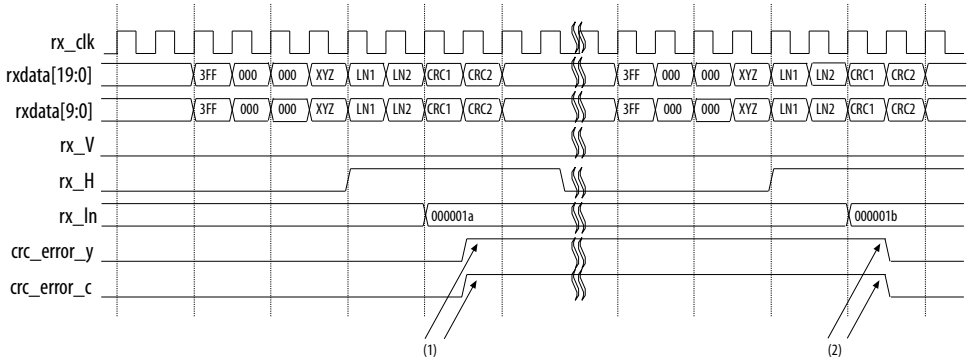


Figure 28. Behavior of crc_error_y and crc_error_c Signals

When a CRC error occurs, the `crc_error_y` or `crc_error_c` signal goes high until the next line. For HD, Dual Link, and 3G Level A, only `crc_error_y[0]` and `crc_error_c[0]` signals are used. For 3G Level B, `crc_error_y[0]` and `crc_error_c[0]` signals are used for link B, and `crc_error_y[1]` and `crc_error_c[1]` signals are used for link A.



(1) The CRC error signals are asserted after the CRC data checked, when the rx_H signal is high. A high CRC signal indicates that there is error in the previous line data. In this case, both Y and C have CRC error.
(2) The CRC error signals are deasserted on the next line, after the CRC data is checked.

Figure 29. Behavior of rx_data_valid Signal—425MA

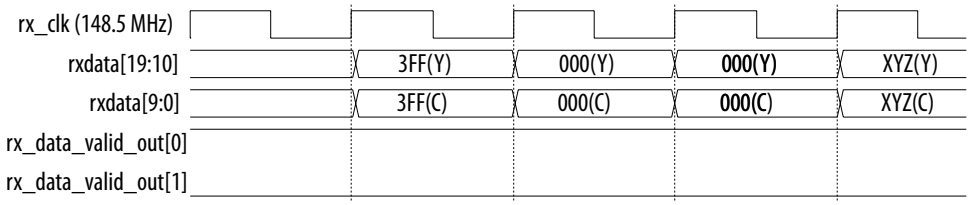


Figure 30. Behavior of rx_data_valid Signal—425MB

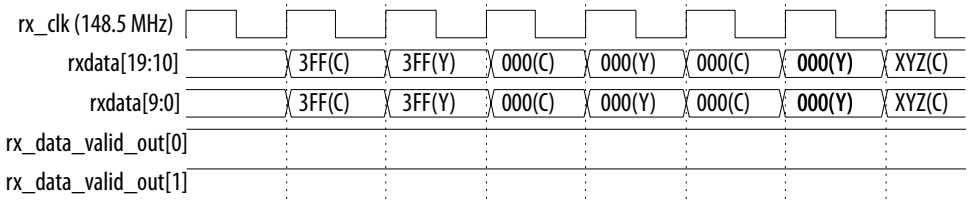


Figure 31. Behavior of rx_trs, rx_xyz, xyz_valid and rx_eav Signals—425MA

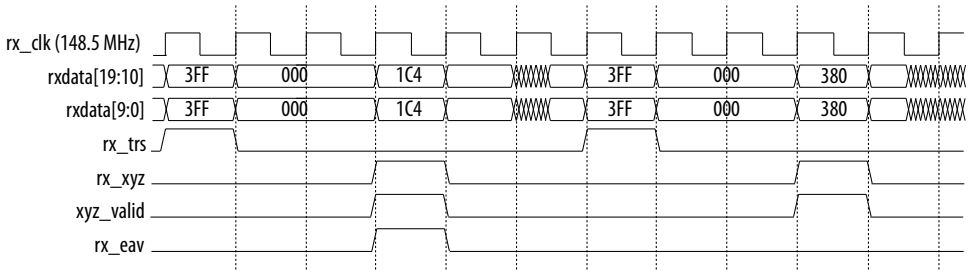
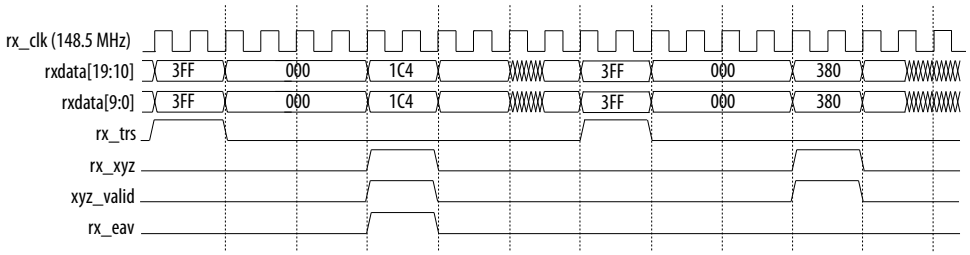


Figure 32. Behavior of rx_trs, rx_xyz, xyz_valid and rx_eav Signals—425MB



4.3. SDI Status Signals

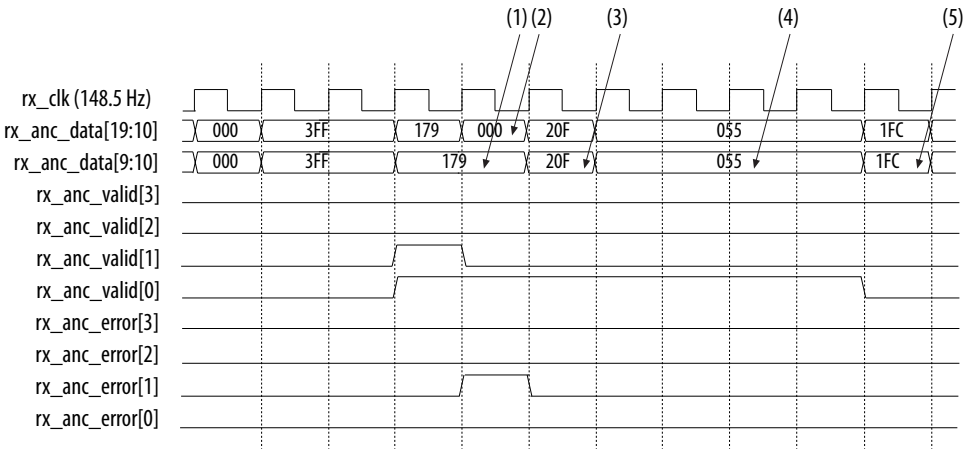
Signal	Width	Direction	Description
rx_anc_data	[(20N−1):0]	Output	Received ancillary data. • SD-SDI: bits 19:10 unused; bits 9:0 Cb, Y, Cr, Y multiplex • HD-SDI: bits 19:10 Y; bits 9:0 C • Dual link: bits 39:30 Y (link B); bits 29:20 C (link B); bits 19:10 Y (link A); bits 9:0 C (link A) • 3G-SDI Level A: bits 19:10 Y; bits 9:0 C • 3G-SDI Level B: bits 19:10 Cb, Y, Cr, Y multiplex (link A); bits 9:0 Cb, Y, Cr, Y multiplex (link B)
rx_anc_error	[3:0]	Output	Ancillary data or checksum error. • SD-SDI: bits 3:1 unused; bits 0 rx_anc_error • HD-SDI: bits 3:2 unused; bit 1 Y; bit 0 C • Dual link: bit 3 Y (link B); bit 2 C (link B); bit 1 Y (link A); bit 0 C (link A) • 3G-SDI Level A: bits 3:2 unused; bit 1 Y; bit 0 C • 3G-SDI Level B: bit 3 Y (link A); bit 2 C (link A); bit 1 Y (link B); bit 0 C (link B)

continued...

Signal	Width	Direction	Description
rx_anc_valid	[3:0]	Output	Ancillary data valid. Asserted to accompany data ID (DID), secondary data ID/data block number (SDID/DBN), data count (DC), and user data words (UDW) on rx_anc_data. • SD-SDI: bits 3:1 unused; bits 0 rx_anc_valid • HD-SDI: bits 3:2 unused; bit 1 Y; bit 0 C • Dual link: bit 3 Y (link B); bit 2 C (link B); bit 1 Y (link A); bit 0 C (link A) • 3G-SDI Level A: bits 3:2 unused; bit 1 Y; bit 0 C • 3G-SDI Level B: bit 3 Y (link A); bit 2 C (link A); bit 1 Y (link B); bit 0 C (link B)
rx_status	[10:0]	Output	This signal is active low for the transceiver-based device families. Receiver status: • rx_status[10] dual link ports aligned • rx_status[9] link B frame locked • rx_status[8] link B TRS locked (six consecutive TRSs with same timing) • rx_status[7] link B alignment locked (a TRS has been spotted and word alignment performed) • rx_status[6] link B receiver in reset • rx_status[5] link B transceiver PLL locked • rx_status[4] link A frame locked • rx_status[3] link A TRS locked (six consecutive TRSs with same timing • rx_status[2] link A alignment locked (a TRS has been spotted and word alignment performed) • rx_status[1] link A receiver in reset • rx_status[0] link A transceiver PLL locked For non HD-SDI dual link versions, only bits [4:0] are active. For transceiver only receiver block in HD-SDI dual link versions, only bits [6:5] and [1:0] are active. This signal indicates lock of the PLL when the transceiver is training from a refclk source. This signal may oscillate when the transceiver is correctly locked to the incoming data in HD-SDI or 3G-SDI modes. In SD-SDI modes, maintain this signal at PLL locked at all times. For rx_status[3] and rx_status[8], the TRS spacing is not required to meet a particular SMPTE standard, but it must be consistent over time for this signal to remain active.
tx_status	[(N– 1):0]	Output	This signal is active low for the transceiver-based device families. Transmitter status, which indicates the transmitter PLL has locked to the tx_serial_refclk signal.

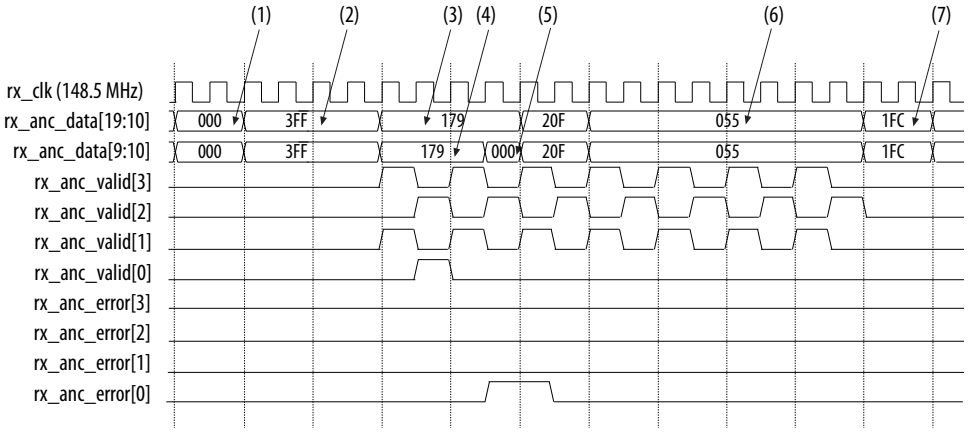
The following figures show the behavior of the rx_anc_data signal.

Figure 33. Behavior of rx_anc_data/valid/error Signals—425MA/HD



Notes:
(1) Sequence starts with Data Indentifier (DID), followed by Secondary Data Indentifier (SDID) or Data Block Number (DBN).
(2) The Y channel goes wrong.
(3) Data Count (DC)
(4) User data word (UDW) up to 255.
(5) Checksum word.

Figure 34. Behavior of rx_anc_data/valid/error Signals—425MB



Notes:
(1) 000 (C), 000 (Y)
(2) 3FF (C), 3FF (C), 3FF (Y)
(3) Sequence starts with Data Indentifier (DID), followed by Secondary Data Indentifier (SDID) or Data Block Number (DBN).
(4) The Y channel of Link B goes wrong.
(5) Data Count (DC) word.
(6) User data word (UDW) up to 255.
(7) Checksum word.

Figure 35. Behavior of rx_anc_data/valid/error Signals Without Error—SD

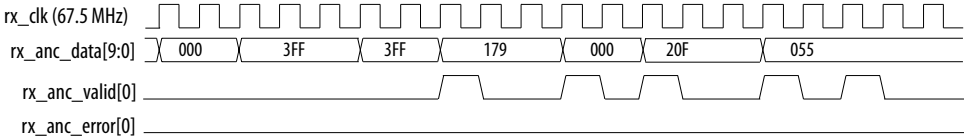
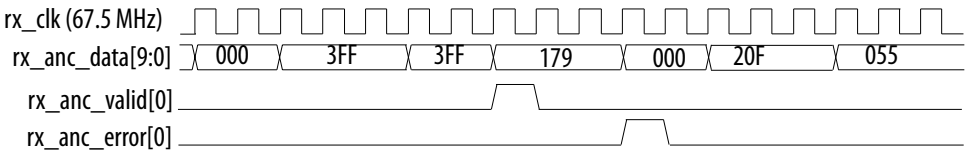


Figure 36. Behavior of rx_anc_data/valid/error Signals With Error—SD



4.4. SDI Transceiver Dynamic Reconfiguration Signals

These signals handle the transceiver dynamic reconfiguration operation.

Table 18. SDI Transceiver Dynamic Reconfiguration Signals

Signal	Direction	Description
SDI_RECONFIG_DONE	Input	Indicates back to the IP core that reconfiguration has finished. This signal is not required for PLL reconfiguration.
SDI_RECONFIG_TOGXB *Note:* Connect this signal directly to a reconfiguration megafunction.	Input	Data input for the embedded transceiver instance. Data width: • For Arria V, Cyclone V, and Stratix V devices = SDI_RECONFIG_TOGXB[(140N-1)]:0] or SDI_RECONFIG_TOGXB[(70N-1)]:0]—if you select the receiver in the interface settings. • For other supported devices = SDI_RECONFIG_TOGXB[3:0] *Note:* SDI transmitters do not require the use of transceiver dynamic reconfiguration. However, to enable the cores to merge into a transceiver quad that has transceiver dynamic reconfiguration enabled, you must connect these ports correctly.
SDI_RECONFIG_CLK	Input	Clock input for the embedded transceiver instance. *Note:* This signal is not applicable for Arria V, Cyclone V, and Stratix V devices. *Note:* SDI transmitters do not require the use of transceiver dynamic reconfiguration. However, to enable the cores to merge into a transceiver quad that has transceiver dynamic reconfiguration enabled, you must connect these ports correctly.
SDI_GXB_POWERDOWN	Input	Powers down and resets circuits in all transceiver instance. *Note:* This signal is not applicable for Arria V, Cyclone V, and Stratix V devices.
SDI_START_RECONFIG	Output	Request from the IP core to start reconfiguration.
SDI_RECONFIG_FROMGXB *Note:* Connect this signal directly to a reconfiguration megafunction.	Output	Data output from the embedded transceiver instance. Data width: • For Arria V, Cyclone V, and Stratix V devices = SDI_RECONFIG_FROMGXB[(92N-1):0] or SDI_RECONFIG_FROMGXB[(46N-1):0]—if you select the receiver in the interface settings. • For other supported devices = SDI_RECONFIG_FROMGXB[(17N-1):0]; SDI_RECONFIG_FROMGXB[16:5] are negligible for Cyclone IV GX devices. *Note:* SDI transmitters do not require the use of transceiver dynamic reconfiguration. However, to enable the cores to merge into a transceiver quad that has transceiver dynamic reconfiguration enabled, you must connect these ports correctly.
RX_STD[1:0]	Output	Receive video standard. 00 = SD-SDI, 01 = HD-SDI, 10 = 3G-SDI.

continued...

Signal	Direction	Description
		The SDI IP core can recover both SMPTE 425M-A and 425M-B formatted streams. The receiver indicates which format it detects by setting the level of the rx_std bus: • rx_std[1:0] = 2'b11 = 425M-A • rx_std[1:0] = 2'b10 = 425M-B
PLL_ARESET	Input	Drives the areset signal on the transceiver PLL to be reconfigured. This signal indicates that the transceiver PLL must be reset. *Note:* The transceivers are available for Cyclone IV GX devices only.
PLL_CONFIGUPDATE	Input	Drives the configupdate signal on the transceiver PLL to be reconfigured. *Note:* The transceivers are available for Cyclone IV GX devices only.
PLL_SCANCLK	Input	Drives the scanclk signal on the transceiver PLL to be reconfigured. *Note:* The transceivers are available for Cyclone IV GX devices only.
PLL_SCANCLKENA	Input	Acts as a clock enable for the scanclk signal on the transceiver PLL to be reconfigured. *Note:* The transceivers are available for Cyclone IV GX devices only.
PLL_SCANDATA	Input	Drives the scandata signal on the transceiver PLL to be reconfigured. This signal holds the scan data input to the transceiver PLL for the dynamically reconfigurable bits. *Note:* The transceivers are available for Cyclone IV GX devices only.
PLL_SCANDONE	Output	Determines when the transceiver PLL is reconfigured. *Note:* The transceivers are available for Cyclone IV GX devices only.
PLL_SCANDATAOUT	Output	This signal holds the transceiver PLL scan data output from the dynamically reconfigurable bits. *Note:* The transceivers are available for Cyclone IV GX devices only.

Note: In the Intel Quartus Prime software version 8.1 and later, the Stratix IV transceivers require receiver buffer calibration through an ALTGX_RECONFIG(transceiver dynamic reconfiguration) controller. The additional RECONFIG port bits are used for receiver buffer calibration. The additional RECONFIG port bits are used for receiver buffer calibration. You must connect these ports to the ALTGX_RECONFIG controller externally.

If you are using Cyclone IV devices, you would require the following additional signals.

Connect these signals directly to the ALTPLL_RECONFIG IP core and expose the signals to the top level when you select the **Use PLL reconfiguration for transceiver dynamic reconfiguration** option in the SDI parameter editor.

Table 19. SDI Transceiver Dynamic Reconfiguration Signals—for Cyclone IV GX Devices

You require rx_std and sdi_start_reconfig signals for PLL reconfiguration.

Signal	Direction	Description
PLL_ARESET	Input	Drives the areset signal on the transceiver PLL to be reconfigured. This signal indicates that the transceiver PLL must be reset.
PLL_CONFIGUPDATE	Input	Drives the configupdate signal on the transceiver PLL to be reconfigured.
PLL_SCANCLK	Input	Drives the scanclk signal on the transceiver PLL to be reconfigured.
PLL_SCANCLKENA	Input	Acts as a clock enable for the scanclk signal on the transceiver PLL to be reconfigured.

continued...

Signal	Direction	Description
PLL_SCANDATA	Input	Drives the scandata signal on the transceiver PLL to be reconfigured. This signal holds the scan data input to the transceiver PLL for the dynamically reconfigurable bits.
PLL_SCANDONE	Output	Determines when the transceiver PLL is reconfigured.
PLL_SCANDATAOUT	Output	This signal holds the transceiver PLL scan data output from the dynamically reconfigurable bits.

5. SDI IP Core User Guide Archives

If an IP core version is not listed, the user guide for the previous IP core version applies.

Intel Quartus Prime Version	User Guide
12.1	Serial Digital Interface (SDI) MegaCore Function User Guide

Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

6. Document Revision History for the SDI IP Core User Guide

Table 20. Document Revision History

Document Version	Intel Quartus Prime Version	Changes
2020.08.20	19.1	• Rebranded as Intel. • Added notification that Intel has discontinued support for this IP core. You may continue to use this IP without support. However, Intel recommends that you use the SDI II Intel FPGA IP for the latest features and continued support. • Removed information about the following devices: — Arria GX — Cyclone, Cyclone II, and Cyclone III — HardCopy III and HardCopy IV — Stratix, Stratix II, and Stratix III Intel no longer supports these devices. • Removed the *SDI Audio IP Cores* chapter from the user guide. Refer to the SDI Audio Intel FPGA IP User Guide for SDI audio information.

Date	Version	Changes
February 2013	12.1	• Updated Table 1–2, Table 1–4, Table 1–5, and Table 1–7 for version 12.1 release. • Updated information on duplex setting in Table 3–10. • Updated information on `rx_video_format` signal in Table 3–12. • Added a note in Table 3–13 to include information about Arria V and Stratix V devices. • Added Table 3–14—transceiver PHY management clock and reset signals. • Updated information on `rx_status` signal in Table 3–18. • Added data width information for `SDI_RECONFIG_TOGXB` and `SDI_RECONFIG_FROMGXB` signals in Table 3–19. • Updated the **Starting channel number** parameter description in Table 3–21. • Added reset sequence information and timing diagram in Figure 3–25.
November 2011	12.0	• Added information about Arria V and Stratix V devices. • Updated Table 1–2, Table 1–5, Table 1–6, and Table 1–7 for version 11.1 release. • Updated *Parameterizing* section to include additional steps to turn on the **Enable TX PLL select for 1/1.000 and 1/1.001 data rate reconfiguration** option. • Updated *Transmitter Clocks* and *Transceiver—Arria GX, Arria II GX, Arria V, Cyclone IV GX, Cyclone V, Stratix II GX, Stratix IV GX, and Stratix V Devices* section and, Table 3–7, Table 3–9, Figure 3–3, and Figure 3–8, to include information about the optional serial reference clock feature.

continued...

Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

Date	Version	Changes
		• Updated Table 3–21 with **Enable TX PLL select for 1/1.000 and 1/1.001 data rate reconfiguration** parameter. • Updated information in the *Transceiver Dynamic Reconfiguration for Dual Standard and Triple Standard Receivers* section. • Updated Table 4–1, Table 4–4, and Table 4–14 to include information about asynchronous and synchcronous modes.
July 2011	11.1	• Added information about accessing transceiver. • Updated Table 3–12 with new signals, `refclk_rate` and `rx_video_format`. • Updated the high-level block diagram of design example for the SDI Audio IP Core to include AES input and output modules. • Updated the SDI Audio IP Core register maps.
December 2010	11.0	• Added two new GUI parameters for SDI MegaCore function: **Enable Spread Spectrum feature** and **Tolerance to consecutive missed EAV**. • Added a chapter on the SDI Audio IP Cores: SDI Audio Embed, Audio Extract, Clocked Audio Input, and Clocked Audio Output MegaCore functions.
July 2010	10.1	• Added information for Cyclone IV devices. • Added a section on transceiver dynamic reconfiguration with PLL reconfiguration mode - Cyclone IV GX. • Added transceiver dynamic reconfiguration signals for PLL reconfiguration in Table 3–18. • Updated Figure 3–29 and Figure 3–30 to include `tx_ln` signal behavior.
November 2009	10.0	• Added Cyclone III LS and Cyclone IV support. • Added a section on *Specify Constraints*. • Updated information on `rst_rx` and `rst_tx` signals in Table 3–15. • Added block diagram for input and output interface signals flow. • Added top-level block diagram for transmitter and receiver.
May 2009	9.1	• Added a section on *Reset Requirement During Reconfiguration*. • Updated information on `txdata`, `tx_ln`, `crc_error_y`, `crc_error_c`, `rx_AP`, `rxdata`, `rx_data_valid_out`, `rx_F`, `rx_H`, `rx_ln`, and `rx_V` signals in Table 3–15. • Updated information on `rx_anc_data`, `rx_anc_error`, `rx_anc_valid`, and `rx_status` signals in Table 3–17.
March 2009	9.0	• Added Arria II GX support. • Added a section on RP168. • Updated information on video formats. • Removed `tx_data_valid_a_bn` signal.
November 2008	9.0	• Added a section on Locking *Algorithm*. • Added new signals and updated existing signal descriptions. • Updated *Appendix A: Constraints*.
May 2008	8.1	• Added Stratix IV support. • Improved receiver lock algorithm. • Updated 425MB support.
October 2007	8.0	• Updated device support. • Updated standards support—3G-SDI now supports *SMPTE425M-B 2006 3Gb/s Signal/Data Serial Interface – Source Image Format Mapping*. • Changed `rx_std` signal description. • Added `tx_data_valid_a_bn` signal.

continued...

Date	Version	Changes
May 2007	7.2	• Updated device support. • Added dual and triple standard information. • Added transceiver dynamic reconfiguration information.
December 2006	7.0	Added support for Cyclone III devices.
December 2006	6.1	Updated for new MegaWizard Plug-In Manager.

A. Constraints

For the SDI IP core to work reliably, you must implement certainIntel Quartus Prime constraints.

- Specify clock characteristics
- Set timing exceptions such as false path, maximum and minimum delays, and multicycle path
- Minimize the timing skew among the paths from I/O pins to the four sampling registers
- Set the oversampling clock that the oversampling interface to 135 MHz uses as an independent clock domain

A.1. Specifying Timing Analyzer Constraints

To ensure your design meets timing and other requirements, you must constrain the design. This section provides the necessary steps to properly constrain your SDI design using the TimeQuest timing analyzer.

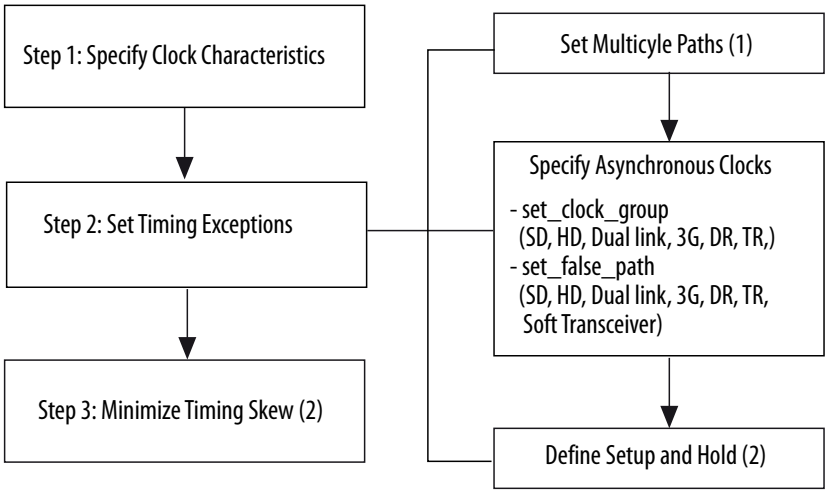
1. Make sure that TimeQuest is specified as the default timing analyzer in the **Timing Analysis Settings** page of the **Settings** dialog box.
2. Compile to create an initial design database before you specify timing constraints for your design. On the Processing menu, click **Start Compilation**.
   A message indicates when compilation is complete.
3. On the Tools menu, click **TimeQuest Timing Analyzer**.
4. Create timing netlist, double-click **Create Timing Netlist** in the **Tasks** pane.
   The timing netlist appears in the Report pane.
5. Specify timing constraints and exceptions. To enter your timing requirements, you can use constraint entry dialog boxes or edit the previously created `.sdc` file.
6. To save your constraints in an `.sdc` file, on the Constraints menu, click **Write SDC File**.

Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

Figure 37. Constraints Design Flow

The figure below shows the flow of the constraint design.



(1) Applicable for SD-SDI only.
(2) Applicable for Soft SERDES only.

Table 21. Step 1: Specify Clock Characteristics

Standard	Clocks	Units
SDI-SD	transceiver_data_rate	270 Mbps
	tx_pclk	27 MHz
	tx_serial_refclk	67.5 MHz
	rx_sd_oversample_clk_in	67.5 MHz
HD-SDI, HD-SDI dual link	transceiver_data_rate	1,485 Mbps
	tx_pclk	74.25 MHz
	tx_serial_refclk	74.25 MHz
	rx_serial_refclk	74.25 MHz
3G-SDI	transceiver_data_rate	2,970 Mbps
	tx_pclk	148.5 MHz
	tx_serial_refclk	148.5 MHz
	rx_serial_refclk	148.5 MHz
DR, TR	transceiver_data_rate	2,970 Mbps
	tx_pclk	148.5 MHz
	tx_serial_refclk	148.5 MHz
	rx_serial_refclk	148.5 MHz
Soft transceiver SDI	rx_sd_refclk_135	135 MHz
	rx_sd_refclk_337	337 MHz

continued...

Standard	Clocks	Units
	rx_sd_refclk_337_90°	337 MHz
	tx_sd_refclk_270	270 MHz
	tx_pclk	27 MHz

Table 22. Step 2: Set Timing Exceptions

Switchline is an internal signal equivalent to the en_switch_reg signal.

Standard	Set Multicycle Paths	set_clock_group	set_false_path (1)	Define Setup and Hold Relationship
SD-SDI	u_format* to u_format	tx_pclk, transmit_pcs0|clkout(gxb_tx_corec lk)	switchline, get_clocks receive_pcs0|clkout (gxb_rxclk)	—
HD-SDI, HD-SDI dual link, 3G-SDI, DR, TR	—	rx_serial_refclk, receive_pcs0|clkout (gxb_rxclk)	switchline, get_clocks receive_pcs0|clkout (gxb_rxclk)	—
		tx_pclk, transmit_pcs0|clkout(gxb_tx_corec lk)	—	
Soft transceiver SDI	—	—	switchline, get_clocks receive_pcs0|clkout (gxb_rxclk)	Setup—1.5 clocks (4.43 ns) from the 337.5-MHz zero-degree clock to the 135-MHz clock
				Hold—zero clocks from the 337.5-MHz clock to the 135-MHz clock

Table 23. Step 3: Minimize the Timing Skew

Standard	Minimize Timing Skew
SD-SDI, HD-SDI, HD-SDI dual link, 3G-SDI, DR, TR	—
Soft transceiver SDI	I/O to sample_a|b|c|d[0] path as short as possible

A.1.1. Constraints for SDI IP Core Using Stratix IV Device

These constraints are specifically used to constraint a duplex SDI IP core.

Specify Clock Characteristics

- SD-SDI (rx_sd_oversample_clk_in = 67.5 MHz, tx_pclk = 27 MHz, tx_serial_refclk = 67.5 MHz)

```
create_clock -name {rx_sd_oversample_clk_in} -period 14.814 -waveform
{ 0.000 7.407 } [get_ports {rx_sd_oversample_clk_in}]
create_clock -name {tx_pclk} -period 14.814 -waveform { 0.000 7.407 }
```

```
[get_ports {tx_pclk}]
create_clock -name {tx_serial_refclk} -period 14.814 -waveform { 0.000
7.407 } [get_ports {tx_serial_refclk}]
```

- HD-SDI, HD-SDI dual link (`rx_serial_refclk` = 74.25 MHz, `tx_pclk` = 74.25 MHz, `tx_serial_refclk` = 74.25 MHz)

```
create_clock -name {rx_serial_refclk} -period 13.468 -waveform { 0.000
6.734 } [get_ports {rx_serial_refclk}]
create_clock -name {tx_pclk} -period 13.468 -waveform { 0.000 6.734 }
[get_ports {tx_pclk}]
create_clock -name {tx_serial_refclk} -period 13.468 -waveform { 0.000
6.734 } [get_ports {tx_serial_refclk}]
```

- 3G-SDI (`rx_serial_refclk` = 148.5 MHz, `tx_pclk` = 148.5 MHz, `tx_serial_refclk` = 148.5 MHz)

```
create_clock -name {rx_serial_refclk} -period 6.734 -waveform { 0.000
3.367 } [get_ports {rx_serial_refclk}]
create_clock -name {tx_pclk} -period 6.734 -waveform { 0.000 3.367 }
[get_ports {tx_pclk}]
create_clock -name {tx_serial_refclk} -period 6.734 -waveform { 0.000
3.367 } [get_ports {tx_serial_refclk}]
```

- Dual standard, triple standard SDI

```
create_clock -name {rx_serial_refclk} -period 6.734 -waveform { 0.000
3.367 } [get_ports {rx_serial_refclk}]
create_clock -name {tx_serial_refclk} -period 6.734 -waveform { 0.000
3.367 } [get_ports {tx_serial_refclk}]
create_clock -name {tx_pclk} -period 6.734 -waveform { 0.000 3.367 }
[get_ports {tx_pclk}]
```

- Soft transceiver SDI

```
create_clock -name {rx_sd_refclk_135} -period 7.407 -waveform { 0.000
3.703 } [get_ports {rx_sd_refclk_135}]
create_clock -name {rx_sd_refclk_337} -period 2.967 -waveform { 0.000
1.484 } [get_ports {rx_sd_refclk_337}]
create_clock -name {rx_sd_refclk_337_90deg} -period 2.967 -waveform { 0.000
1.484 } [get_ports {rx_sd_refclk_337_90deg}]
create_clock -name {tx_sd_refclk_270} -period 3.703 -waveform { 0.000
1.852 } [get_ports {tx_sd_refclk_270}]
create_clock -name {tx_pclk} -period 37.037 -waveform { 0.000 18.519 }
[get_ports {tx_pclk}]
```

Set Multicycle Paths

In some device families and speed grades, timing violations may occur in the format block of the SDI IP core. For SD-SDI, these violations are multicycle, and you can fix by applying the following constraints to your design.

Note: These constraints apply only to SD-SDIs; the violations are single-cycle paths in other video standards.

```
set_multicycle_path -setup -end -from [get_keepers
{sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[0].u_txrx_port|sdi_format:format_gen.u_format|
*}] -to [get_keepers {sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[0].u_txrx_port|sdi_format:format_gen.u_format|
*}] 2
set_multicycle_path -hold -end -from [get_keepers
{sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[0].u_txrx_port|sdi_format:format_gen.u_format|
```

```
*}] -to [get_keepers {sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[0].u_txrx_port|sdi_format:format_gen.u_format|
*}] 1
```

Specify Clocks that are Exclusive or Asynchronous

The SDI IP core may show timing violations in slower speed grade devices. These paths are not required to have fast timing, so you can use the following constraints to remove these timing paths. You can use the command `set_clock_groups` or `set_false_path`.

Note: The following SDC commands apply only for duplex core and Stratix IV devices; you must use the constraint entry dialog boxes to constrain the separate receiver or transmitter core and other device families.

- SD-SDI

```
set_clock_groups -exclusive -group [get_clocks {tx_pclk}] -group [get_clocks
{sdi_megacore_top_inst|sdi_txrx_port_gen[0].u_txrx_port|
gen_duplex_alt4gxb.u_gxb|alt4gxb_component|auto_generated|transmit_pcs0|
clkout}]
set_false_path -from [get_keepers {sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[0].u_txrx_port|switchline}] -to [get_clocks
{sdi_megacore_top_inst|sdi_txrx_port_gen[0].u_txrx_port|
gen_duplex_alt4gxb.u_gxb|alt4gxb_component|auto_generated|receive_pcs0|
clkout}]
```

- HD-SDI, 3G-SDI, dual standard, triple standard SDI

```
set_clock_groups -exclusive -group [get_clocks {rx_serial_refclk}] -group
[get_clocks {sdi_megacore_top_inst|sdi_txrx_port_gen[0].u_txrx_port|
gen_duplex_alt4gxb.u_gxb|alt4gxb_component|auto_generated|receive_pcs0|
clkout}]
set_clock_groups -exclusive -group [get_clocks {tx_pclk}] -group [get_clocks
{sdi_megacore_top_inst|sdi_txrx_port_gen[0].u_txrx_port|
gen_duplex_alt4gxb.u_gxb|alt4gxb_component|auto_generated|transmit_pcs0|
clkout}]
set_false_path -from [get_keepers {sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[0].u_txrx_port|switchline}] -to [get_clocks
{sdi_megacore_top_inst|sdi_txrx_port_gen[0].u_txrx_port|
gen_duplex_alt4gxb.u_gxb|alt4gxb_component|auto_generated|receive_pcs0|
clkout}]
```

- HD-SDI dual link (for the additional channel)

```
set_clock_groups -exclusive -group [get_clocks {rx_serial_refclk}] -group
[get_clocks {sdi_megacore_top_inst|sdi_txrx_port_gen[1].u_txrx_port|
gen_duplex_alt4gxb.u_gxb|alt4gxb_component|auto_generated|receive_pcs0|
clkout}]
set_false_path -from [get_keepers {sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[0].u_txrx_port|switchline}] -to [get_clocks
{sdi_megacore_top_inst|sdi_txrx_port_gen[0].u_txrx_port|
gen_duplex_alt4gxb.u_gxb|alt4gxb_component|auto_generated|receive_pcs0|
clkout}]
set_clock_groups -exclusive -group [get_clocks {tx_pclk}] -group [get_clocks
{sdi_megacore_top_inst|sdi_txrx_port_gen[1].u_txrx_port|
gen_duplex_alt4gxb.u_gxb|alt4gxb_component|auto_generated|transmit_pcs0|
clkout}]
set_false_path -from [get_keepers {sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[1].u_txrx_port|switchline}] -to [get_clocks
{sdi_megacore_top_inst|sdi_txrx_port_gen[1].u_txrx_port|
gen_duplex_alt4gxb.u_gxb|alt4gxb_component|auto_generated|receive_pcs0|
clkout}]
```

Define the Setup and Hold Relationship between 135-MHz Clocks and 337.5-MHz Zero-degree Clocks

These constraints apply only to soft transceiver SDI.

- Setup—1.5 clocks (4.43 ns) from the 337.5-MHz zero-degree clock to the 135-MHz clock
- Hold—zero clocks from the 337.5-MHz clock to the 135-MHz clock

  Use the `set_min_delay` command to specify an absolute minimum delay for a given path.

  ```
  set_min_delay -from  [get_clocks {rx_sd_refclk_337}]  -to  [get_clocks
  {rx_sd_refclk_135}] 0.000
  ```

  Use the `set_max_delay` command to specify an absolute maximum delay for a given path.

  ```
  set_max_delay -from [get_clocks {rx_sd_refclk_337}] -to [get_clocks
  {rx_sd_refclk_135}] 4.430
  ```

Minimize Timing Skew

You must minimize the timing skew among the paths from I/O pins to the four sampling registers.

- `sample_a[0]`
- `sample_b[0]`
- `sample_c[0]`
- `sample_d[0]`

To minimize the timing skew:

- Manually place the sampling registers close to each other and to the serial input pin.
- Because these four registers use four different clock domains, place two of the four registers in one LAB and the other two in another LAB.
- Then, place the two chosen LABs within the same row regardless of the placement of the serial input.
- Finally, do not place the four sampling registers at the immediate rows or columns next to the I/O, but at the second row or column next to the I/O bank. This location allows faster inter-LAB interconnections between I/O banks and their immediate rows or columns compared to core interconnection.

The following code is an example of a constraint, which you can set using the Intel Quartus Prime Assignment Editor:

```
set_location_assignment PIN_99 -to sdi_rx
set_location_assignment LC_X32_Y17_N0 -to
"sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[0].u_txrx_port|
soft_serdes_rx:rx_soft_serdes_gen.soft_serdes_rx_inst|serdes_s2p:u_s2p|
sample_a[0]"
set_location_assignment LC_X33_Y17_N0 -to
"sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[0].u_txrx_port|
soft_serdes_rx:rx_soft_serdes_gen.soft_serdes_rx_inst|serdes_s2p:u_s2p|
sample_b[0]"
set_location_assignment LC_X32_Y17_N1 -to
```

```
"sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[0].u_txrx_port|
soft_serdes_rx:rx_soft_serdes_gen.soft_serdes_rx_inst|serdes_s2p:u_s2p|
sample_c[0]"
set_location_assignment LC_X33_Y17_N1 -to
"sdi_megacore_top:sdi_megacore_top_inst|
sdi_txrx_port:sdi_txrx_port_gen[0].u_txrx_port|
soft_serdes_rx:rx_soft_serdes_gen.soft_serdes_rx_inst|serdes_s2p:u_s2p|
sample_d[0]"
```

A.2. Constraints for the SDI Soft Transceiver

These constraints apply to all supported device families that are configured to use a soft transceiver for their receivers.

Define the following setup and hold relationship between the 135-MHz clocks and the 337.5-MHz zero-degree clocks:

- Setup—1.5 clocks (4.43 ns) from the 337.5-MHz zero-degree clock to the 135-MHz clock
- Hold—zero clocks from the 337.5-MHz clock to the 135-MHz clock

If you choose to include the PLLs inside the IP core, modify the following constraints and apply them to your design. Alternatively, apply similar constraints to the clocks connected to the `rx_sd_refclk_337` and `rx_sd_refclk_135` signals on your SDI IP core.

Classic Timing Analyzer

Use the following constraints for the Classic timing analyzer:

```
set_instance_assignment -name SETUP_RELATIONSHIP "4.43 ns" -from
“<your_megacore:your_megacore_inst>|sdi_megacore_top:sdi_megacore_top_inst|
sdi_clocks:u_sdi_clocks|stratix_c2_pll_sclk:u_rx_pll|altpll:altpll_component|
_clk0" -to "<your_megacore:your_megacore_inst>|
sdi_megacore_top:sdi_megacore_top_inst|sdi_clocks:u_sdi_clocks|
stratix_c2_pll_sclk:u_rx_pll|altpll:altpll_component|_clk2"
set_instance_assignment -name HOLD_RELATIONSHIP "0 ns" -from
"<your_megacore:your_megacore_inst>|sdi_megacore_top:sdi_megacore_top_inst|
sdi_clocks:u_sdi_clocks|stratix_c2_pll_sclk:u_rx_pll|altpll:altpll_component|
_clk0" -to "<your_megacore:your_megacore_inst>|
sdi_megacore_top:sdi_megacore_top_inst|sdi_clocks:u_sdi_clocks|
stratix_c2_pll_sclk:u_rx_pll|altpll:altpll_component|_clk2"
```

TimeQuest Timing Analyzer

Use the following constraints for the TimeQuest timing analyzer:

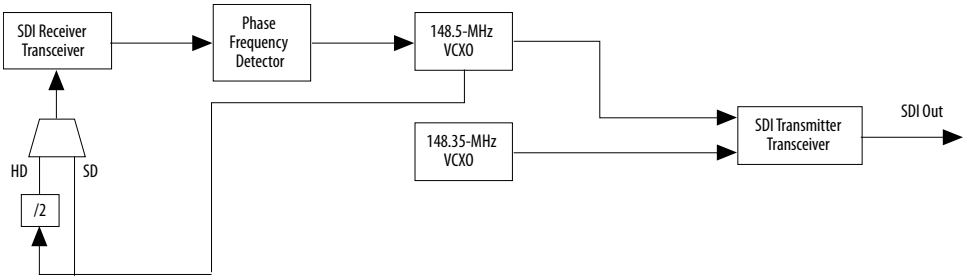
```
set_max_delay 4.43 -from {<your_megacore:your_megacore_inst>|
sdi_megacore_top:sdi_megacore_top_inst|sdi_clocks:u_sdi_clocks|
stratix_c2_pll_sclk:u_rx_pll|altpll:altpll_component|_clk0} -to
{<your_megacore:your_megacore_inst>|sdi_megacore_top:sdi_megacore_top_inst|
sdi_clocks:u_sdi_clocks|stratix_c2_pll_sclk:u_rx_pll|altpll:altpll_component|
_clk2}
set_min_delay 0 -from { <your_megacore:your_megacore_inst>|
sdi_megacore_top:sdi_megacore_top_inst|sdi_clocks:u_sdi_clocks|
stratix_c2_pll_sclk:u_rx_pll|altpll:altpll_component|_clk0} -to
{<your_megacore:your_megacore_inst>|sdi_megacore_top:sdi_megacore_top_inst|
sdi_clocks:u_sdi_clocks|stratix_c2_pll_sclk:u_rx_pll|altpll:altpll_component|
_clk2}
```

B. Clocking

B.1. Clocking Versions

Figure 38. Version 11.1 and Later International Clocks

The figure below shows how you must clock the transceivers for SDI cores with international clocking. The clocking complies to both American and European standards.



Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

C. Receive and Retransmit

You cannot reuse the HD and 3G recovered clocks for transmitting because the transmitter jitter is then entirely dependent on the input jitter and jitter transfer function.

The general recommended approach to system locking with the SDI IP core is to use a voltage-controlled crystal oscillator (VCXO) external to the device. The VCXO must be locked to the receiver clock out of the SDI IP core. The SDI IP core then uses the clean VCXO output as the transmit clock.

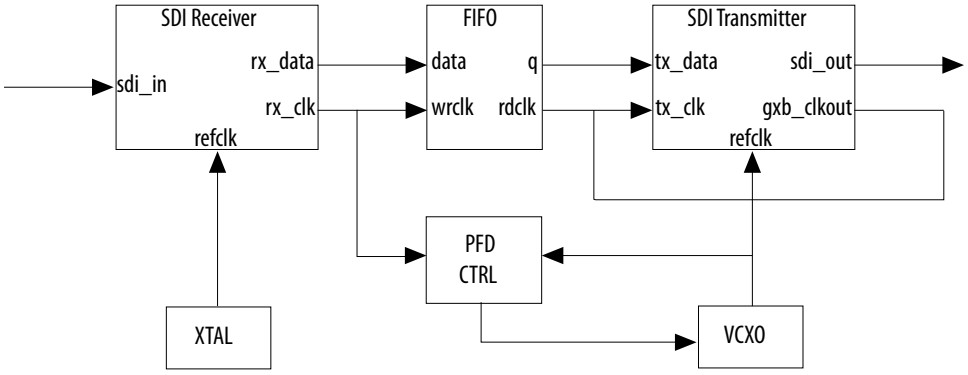
C.1. Loopback FIFO Buffer

For more efficient transmission, place a FIFO or buffer between the receiver clock domain logic and the transmit clock domain logic.

The decoded receiver data connects to the transmitter input through a FIFO buffer. When the receiver is locked, the logic writes the receiver data to the FIFO buffer. When the FIFO is half full, the transmitter starts reading, encoding and transmitting the data.

Figure 39. Receive and Retransmit Clock Scheme

The figure shows the clocking scheme of the received and retransmitted data.



Intel Corporation. All rights reserved. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Altera:

IP-SDI