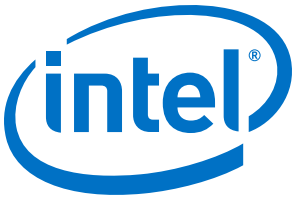


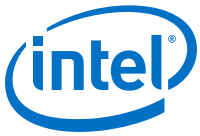
JESD204B Intel FPGA IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **18.0**

Subscribe
Send Feedback

UG-01142 | 2018.05.07
Latest document on the web: **PDF** | **HTML**

Contents



1. JESD204B IP Core Quick Reference

The JESD204B Intel FPGA IP is a high-speed point-to-point serial interface intellectual property (IP).

Note: For system requirements and installation instructions, refer to *Intel FPGA Software Installation & Licensing*.

Table 1. Brief Information About the JESD204B IP Core

Item		Description
Release Information	Version	18.0
	Release Date	May 2018
	Ordering Code	IP-JESD204B
	Product ID	0116
	Vendor ID	6AF7
IP Core Information	Protocol Features	• Joint Electron Device Engineering Council (JEDEC) JESD204B.01, 2012 standard release specification • Device subclass: — Subclass 0—Backwards compatible to JESD204A. — Subclass 1—Uses *SYSREF* signal to support deterministic latency. — Subclass 2—Uses *SYNC_N* detection to support deterministic latency.
IP Core Information	Core Features	• Run-time configuration of parameters L,M, and F • Data rates up to 12.5 gigabits per second (Gbps)—per JESD204B specification • Data rates of up to 16.0 Gbps—not certified per JESD204B specification (uncharacterized support) • Single or multiple lanes (up to 8 lanes per link) • Serial lane alignment and monitoring • Lane synchronization • Modular design that supports multidevice synchronization • MAC and PHY partitioning • Deterministic latency support • 8B/10B encoding • Scrambling/Descrambling • Avalon® Streaming (Avalon-ST) interface for transmit and receive datapaths • Avalon Memory-Mapped (Avalon-MM) interface for Configuration and Status registers (CSR) • Dynamic generation of simulation testbench

continued...

Item		Description
	Typical Application	• Wireless communication equipment • Broadcast equipment • Military equipment • Medical equipment • Test and measurement equipment
	Device Family Support	• Intel Cyclone® 10 GX FPGA devices • Intel Stratix® 10 FPGA devices • Intel Arria® 10 FPGA devices • Stratix V FPGA devices • Arria V FPGA devices • Arria V GZ FPGA devices • Cyclone V FPGA devices
	Design Tools	• Platform Designer parameter editor in the Intel Quartus® Prime software for design creation and compilation • Timing Analyzer in the Intel Quartus Prime software for timing analysis • ModelSim-Intel FPGA, Aldec Riviera-PRO*, Synopsys VCS/VCS MX, Cadence NCSim, and Cadence Xcelium* Parallel simulator software for design simulation or synthesis

Related Information

- Design Examples for JESD204B IP Core User Guide
  Provides information about design examples for Arria V, Cyclone V, Stratix V, and Intel Arria 10 devices using Intel Quartus Prime Standard Edition software.
- JESD204B Intel Arria 10 FPGA IP Design Example User Guide
- JESD204B Intel Stratix 10 FPGA IP Design Example User Guide
- JESD204B Intel Cyclone 10 GX FPGA IP Design Example User Guide
- Intel FPGA Software Installation and Licensing
- What's New in Intel FPGA IP
- JESD204B IP Core Release Notes
- Errata for JESD204B IP Core in the Knowledge Base
- AN803: Implementing ADC-Intel Arria 10 Multi-Link Design with JESD204B RX IP Core
- AN804: Implementing ADC-Intel Stratix 10 Multi-Link Design with JESD204B RX IP Core
- JESD204B Intel FPGA IP Document Archives on page 115
  Provides a list of user guides for previous versions of the JESD204B IP core.

2. About the JESD204B

The JESD204B Intel FPGA IP is a high-speed point-to-point serial interface for digital-to-analog (DAC) or analog-to-digital (ADC) converters to transfer data to FPGA devices. This unidirectional serial interface runs at a maximum data rate of 16.0 Gbps. This protocol offers higher bandwidth, low I/O count and supports scalability in both number of lanes and data rates. The JESD204B Intel FPGA IP addresses multi-device synchronization by introducing Subclass 1 and Subclass 2 to achieve deterministic latency.

Note: The full product name, JESD204B Intel FPGA IP, is shortened to JESD204B IP core in this document.

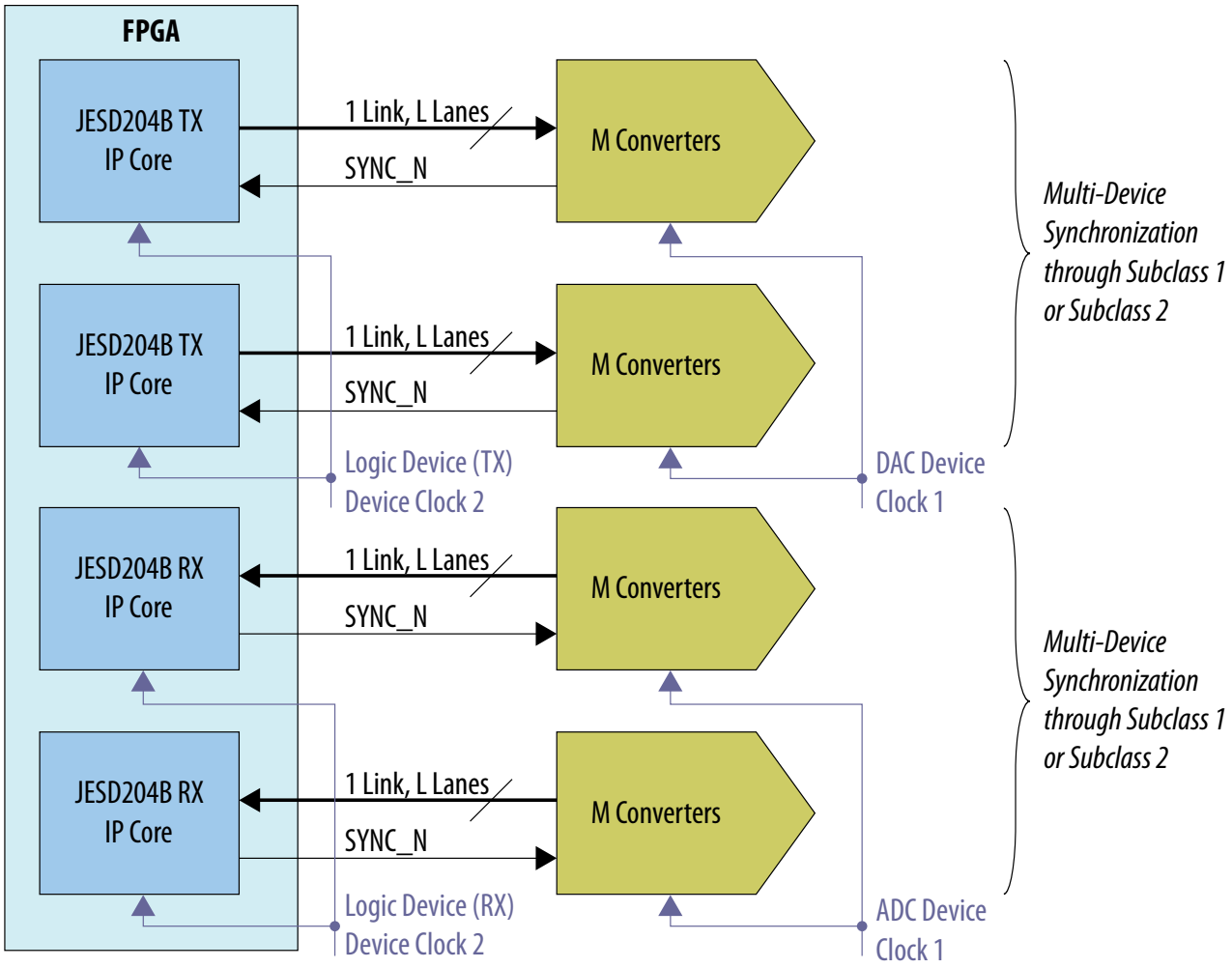
The JESD204B IP core incorporates:

- Media access control (MAC)—data link layer (DLL) block that controls the link states and character replacement.
- Physical layer (PHY)—physical coding sublayer (PCS) and physical media attachment (PMA) block.

The JESD204B IP core does not incorporate the Transport Layer (TL) that controls the frame assembly and disassembly. The TL and test components are provided as part of a design example component where you can customize the design for different converter devices.

Figure 1. Typical System Application for JESD204B IP Core

The JESD204B IP core uses the Avalon-ST source and sink interfaces, with unidirectional flow of data, to transmit and receive data on the FPGA fabric interface.



Key features of the JESD204B IP core:

- Data rate of up to 16.0 Gbps (characterization up to 12.5G)
- Run-time JESD204B parameter configuration (L, M, F, S, N, K, CS, CF)
- MAC and PHY partitioning for portability
- Subclass 0 mode for backward compatibility to JESD204A
- Subclass 1 mode for deterministic latency support (using *SYSREF*) between the ADC/DAC and logic device
- Subclass 2 mode for deterministic latency support (using *SYNC_N*) between the ADC/DAC and logic device
- Multi-device synchronization

Related Information

- V-Series Transceiver PHY User Guide
- Intel Arria 10 Transceiver PHY User Guide
- Intel Cyclone 10 GX Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide
- Intel Stratix 10 Device Datasheet
- Intel Arria 10 Device Datasheet

2.1. Datapath Modes

The JESD204B IP core supports TX-only, RX-only, and Duplex (TX and RX) mode. The IP core is a unidirectional protocol where interfacing to ADC utilizes the transceiver RX path and interfacing to DAC utilizes the transceiver TX path.

The JESD204B IP core generates a single link with a single lane and up to a maximum of 8 lanes. If there are two ADC links that need to be synchronized, you have to generate two JESD204B IP cores and then manage the deterministic latency and synchronization signals, like *SYSREF* and *SYNC_N*, at your custom wrapper level.

The JESD204B IP core supports duplex mode only if the LMF configuration for ADC (RX) is the same as DAC (TX) and with the same data rate. This use case is mainly for prototyping with internal serial loopback mode. This is because typically as a unidirectional protocol, the LMF configuration of converter devices for both DAC and ADC are not identical.

2.2. IP Core Variation

The JESD204B IP core has three core variations:

- JESD204B MAC only
- JESD204B PHY only
- JESD204B MAC and PHY

In a subsystem where there are multiple ADC and DAC converters, you need to use the Intel Quartus Prime software to merge the transceivers and group them into the transceiver architecture. For example, to create two instances of the JESD204B TX IP core with four lanes each and four instances of the JESD204 RX IP core with two lanes each, you can apply one of the following options:

- MAC and PHY option
  1. Generate JESD204B TX IP core with four lanes and JESD204B RX IP core with two lanes.
  2. Instantiate the desired components.
  3. Use the Intel Quartus Prime software to merge the PHY lanes.
- MAC only and PHY only option—based on the configuration above, there are a total of eight lanes in duplex mode.
  1. Generate the JESD204B Duplex PHY with a total of eight lanes. (TX skew is reduced in this configuration as the channels are bonded).
  2. Generate the JESD204B TX MAC with four lanes and instantiate it two times.
  3. Generate the JESD204B RX MAC with two lanes and instantiate it four times.
  4. Create a wrapper to connect the JESD204B TX MAC and RX MAC with the JESD204B Duplex PHY.

Note: If the data rate for TX and RX is different, the transceiver does not allow duplex mode to generate a duplex PHY. In this case, you have to generate a RX-only PHY on the RX data rate and a TX-only PHY on the TX data rate.

2.3. JESD204B IP Core Configuration

Table 3. JESD204B IP Core Configuration

Symbol	Description	Value
L	Number of lanes per converter device	1-8
M	Number of converters per device	1-256
F	Number of octets per frame	1, 2, 4-256
S	Number of transmitted samples per converter per frame	1-32
N	Number of conversion bits per converter	1-32
N'	Number of transmitted bits per sample (JESD204 word size, which is in nibble group)	1-32
K	Number of frames per multiframe	17/F ≤ K ≤ 32 ; 1-32
CS	Number of control bits per conversion sample	0-3
CF	Number of control words per frame clock period per link	0-32
HD	High Density user data format	0 or 1
LMFC	Local multiframe clock	(F × K /4) link clock counts (1)

(1) The value of F x K must be divisible by 4.

2.3.1. Run-Time Configuration

The JESD204B IP core allows run-time configuration of LMF parameters in all supported devices except for Intel Stratix 10. For Intel Stratix 10 devices, the JESD204B IP core must be parameterized according to your target converter device with the IP configurations shown in *JESD204B Configurations Tab* of Table 13 on page 35

Note: For Intel Stratix 10 devices, run-time access for certain registers have been disabled. Refer to the TX and RX register map for more information.

The most critical parameters that must be set correctly during IP generation are the L and F parameters. Parameter L denotes the maximum lanes supported while parameter F denotes the size of the deskew buffer needed for deterministic latency. The hardware generates during parameterization, which means that run-time programmability can only fall back from the parameterized and generated hardware, but not beyond the parameterized IP core.

You can use run-time configuration for prototyping or evaluating the performance of converter devices with various LMF configurations. However, in actual production,Intel recommends that you generate the JESD204B IP core with the intended LMF to get an optimized gate count.

For example, if a converter device supports LMF = 442 and LMF = 222, to check the performance for both configurations, you need to generate the JESD204B IP core with maximum F and L, which is L = 4 and F = 2. During operation, you can use the fall back configuration to disable the lanes that are not used in LMF = 222 mode. You must ensure that other JESD204B configurations like M, N, S, CS, CF, and HD do not violate the parameter F setting. You can access the Configuration and Status Register (CSR) space to modify other configurations such as:

- K (multiframe)
- device and lane IDs
- enable or disable scrambler
- enable or disable character replacement

F Parameter

This parameter indicates how many octets per frame per lane that the JESD204B link is operating in. You must set the F parameter according to the JESD204B IP Specification for a correct data mapping.

To support the High Density (HD) data format, the JESD204B IP core tracks the start of frame and end of frame because F can be either an odd or even number. The start of frame and start of multiframe wrap around the 32-bits data width architecture. The RX IP core outputs the start of frame (`sof[3:0]`) and start of multiframe (`somf[3:0]`), which act as markers, using the Avalon-ST data stream. Based on these markers, the transport layer build the frames.

In a simpler system where the HD data format is set to 0, the F will always be 1, 2, 4, 6, 8, and so forth. This simplifies the transport layer design, so you do not need to use the `sof[3:0]` and `somf[3:0]` markers.

Related Information

- JESD204B RX Address Map and Register Definitions

- JESD204B TX Address Map and Register Definitions

2.4. Channel Bonding

The JESD204B IP core supports channel bonding—bonded (PMA bonding for Intel Stratix 10, Intel Arria 10, and Intel Cyclone 10 GX) and non-bonded modes.

The channel bonding mode that you select may contribute to the transmitter channel-to-channel skew. A bonded transmitter datapath clocking provides low channel-to-channel skew as compared to non-bonded channel configurations.

For Intel Stratix 10, Intel Arria 10, and Intel Cyclone 10 GX devices, refer to PMA Bonding chapter of the respective *Transceiver PHY User Guide* , about how to connect the ATX PLL and fPLL in bonded configuration and non-bonded configuration. For the non-bonded configuration, refer to *Implementing Multi-Channel xN Non-Bonded Configuration*. For bonded configuration, refer to *Implementing x6/xN Bonding Mode*.

- In PHY-only mode, you can generate up to 32 channels, provided that the channels are on the same side. In MAC and PHY integrated mode, you can generate up to 8 channels.
- In bonded channel configuration, the lower transceiver clock skew for all channels result in a lower channel-to-channel skew.
  - For Stratix V, Arria V, and Cyclone V devices, you must use contiguous channels when you select bonded mode. The JESD204B IP core automatically selects between ×6, ×N or feedback compensation (fb_compensation) bonding depending on the number of transceiver channels you set.
  - For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, you do not have to place the channels in bonded group contiguously. Refer to Table 5 on page 12 for the clock network selection. Refer to *Channel Bonding* section of the respective *Transceiver PHY User Guide* for more information about PMA Bonding.
- In non-bonded channel configuration, the transceiver clock skew is higher and latency is unequal in the transmitter phase compensation FIFO for each channel. This may result in a higher channel-to-channel skew.

Table 4. Maximum Number of Lanes (L) Supported in Bonded and Non-Bonded Mode

Device Family	Core Variation	Bonding Mode Configuration	Maximum Number of Lanes (L)
Intel Stratix 10 Intel Arria 10 Intel Cyclone 10 GX Stratix V Arria V GZ Cyclone V	PHY only	Bonded	32 (2)
		Non-bonded	32 (2)
	MAC and PHY	Bonded	8
		Non-bonded	8
Arria V	PHY only	Bonded	32 (2)

continued...

(2) The maximum lanes listed here is for configuration simplicity. Refer to the *Intel FPGA Transceiver PHY User Guide* for the actual number of channels supported.

Device Family	Core Variation	Bonding Mode Configuration	Maximum Number of Lanes (L)
		Non-bonded	32 (2)
	MAC and PHY	Bonded	6
		Non-bonded	8

Table 5. Clock Network Selection for Bonded Mode

Device Family	L ≤ 6	L > 6
Intel Stratix 10 Intel Arria 10 Intel Cyclone 10 GX	×6	×N (3)
Stratix V	×6	feedback compensation
Arria V	×N	×N
Arria V GZ	×6	feedback compensation
Cyclone V	×N	×N

Related Information

- V-Series Transceiver PHY User Guide
- Intel Arria 10 Transceiver PHY User Guide
- Intel Cyclone 10 GX Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide
- Intel Stratix 10 Device Datasheet
- Intel Arria 10 Device Datasheet

(3) Bonded mode is not supported for data rate > 15 Gbps. Refer to the respective datasheet for the maximum data rate and channel span supported by the ×N clock network and the transceiver power supply operating condition for your device.

2.5. Performance and Resource Utilization

Table 6. JESD204B IP Core FPGA Performance

Device Family	PMA Speed Grade	FPGA Fabric Speed Grade	Data Rate		Link Clock F_{MAX} (MHz)
			Enable Hard PCS (Gbps)	Enable Soft PCS (Gbps) [4]	
Intel Stratix 10	1	1	2.0 to 12.0	2.0 to 16.0[6]	data_rate/40
		2	2.0 to 12.0	2.0 to 13.5	data_rate/40
	2	1	2.0 to 9.83	2.0 to 16.0[6]	data_rate/40
		2	2.0 to 9.83	2.0 to 13.5	data_rate/40
	3	1	2.0 to 9.83	2.0 to 16.0[6]	data_rate/40
		2	2.0 to 9.83	2.0 to 13.5	data_rate/40
		3	2.0 to 9.83	2.0 to 12.5	data_rate/40
Intel Arria 10	1	1	2.0 to 12.0	2.0 to 15.0 [6][5]	data rate/40
	2	1	2.0 to 12.0	2.0 to 15.0 [6] [5]	data rate/40
	2	2	2.0 to 9.83	2.0 to 15.0 [6] [5]	data rate/40
	3	1	2.0 to 12.0	2.0 to 14.2 [6] [7]	data rate/40
	3	2	2.0 to 9.83	2.0 to 14.2 [6] [8]	data rate/40
	4	3	2.0 to 8.83	2.0 to 12.5[9]	data rate/40

continued...

(4) Select `Enable Soft PCS` to achieve maximum data rate. For the TX IP core, enabling soft PCS incurs an additional 3–8% increase in resource utilization. For the RX IP core, enabling soft PCS incurs an additional 10–20% increase in resource utilization.

(5) When using Soft PCS mode at 15.0 Gbps, the timing margin is very limited. You are advised to enable high fitter effort, register duplication, and register retiming to improve timing performance.

(6) Refer to the Intel Arria 10 and Intel Stratix 10 Device Datasheet for the maximum data rate supported across transceiver speed grades and transceiver power supply operating conditions.

(7) For Intel Arria 10 GX 160, SX 160, GX 220 and SX 220 devices, the supported data rate is up to 12.288 Gbps.

(8) For Intel Arria 10 GX 160, SX 160, GX 220 and SX 220 devices, the supported data rate is 11.0 Gbps.

Device Family	PMA Speed Grade	FPGA Fabric Speed Grade	Data Rate		Link Clock F_{MAX} (MHz)
			Enable Hard PCS (Gbps)	Enable Soft PCS (Gbps) (4)	
Intel Cyclone 10 GX	*<Any supported speed grade>*	*<Any supported speed grade>*	2.0 to 6.25	2.0 to 6.25	data rate/40
Stratix V	1	1 or 2	2.0 to 12.2	2.0 to 12.5	data rate/40
	2	1 or 2	2.0 to 12.2	2.0 to 12.5	data rate/40
	2	3	2.0 to 9.8	2.0 to 12.5 (10)	data rate/40
	3	1, 2, 3, or 4	2.0 to 8.5	2.0 to 8.5	data rate/40
Arria V GX/SX	*<Any supported speed grade>*	*<Any supported speed grade>*	1.0 to 6.55	— (11)	data rate/40
Arria V GT/ST	*<Any supported speed grade>*	*<Any supported speed grade>*	1.0 to 6.55	4.0 to 7.5 (PMA direct) (11)	data rate/40
Arria V GZ	2	3	2.0 to 9.9	— (11)	data rate/40
	3	4	2.0 to 8.8	— (11)	data rate/40
Cyclone V	5	*<Any supported speed grade>*	1.0 to 5.0	—	data rate/40
	6	6 or 7	1.0 to 3.125	—	data rate/40

The following table lists the resources and expected performance of the JESD204B IP core. These results are obtained using the Intel Quartus Prime software targeting the following Intel FPGA devices:

- Cyclone V: 5CGTFD9E5F31I7
- Arria V: 5AGXFB3H4F35C5
- Arria V GZ: 5AGZME5K2F40C3
- Intel Arria 10: 10AX115H2F34I2SGES

(4) Select `Enable Soft PCS` to achieve maximum data rate. For the TX IP core, enabling soft PCS incurs an additional 3–8% increase in resource utilization. For the RX IP core, enabling soft PCS incurs an additional 10–20% increase in resource utilization.

(9) For Intel Arria 10 GX 160, SX 160, GX 220 and SX 220 devices, the supported data rate is 10.0 Gbps.

(10) When using Soft PCS mode at 12.5 Gbps, the timing margin is very limited. You are advised to enable high fitter effort, register duplication, and register retiming to improve timing performance.

(11) Enabling Soft PCS does not increase the data rate for the device family and speed grade. You are recommended to select the *Enable Hard PCS* option.

- Stratix V: 5SGXEA7H3F35C3
- Intel Stratix 10: 1SG280LN3F43E3VG
- Intel Cyclone 10 GX: 10CX105YF672I6G

All the variations for resource utilization are configured with the following parameter settings:

Table 7. Parameter Settings To Obtain the Resource Utilization Data

Parameter	Setting
JESD204B Wrapper	Base and PHY
JESD204B Subclass	1
Data Rate	5 Gbps
PCS Option	Enabled Hard PCS
PLL Type	• ATX (for 10 series devices) • CMU (for V series devices)
Bonding Mode	Non-bonded
Reference Clock Frequency	125.0 MHz
Octets per frame (F)	1
Enable Scrambler (SCR)	Off
Enable Error Code Correction (ECC_EN)	Off

Table 8. JESD204B IP Core Resource Utilization

The numbers of ALMs and logic registers in this table are rounded up to the nearest 10.

Note: The resource utilization data are extracted from a full design which includes the Intel FPGA Transceiver PHY Reset Controller IP core. Thus, the actual resource utilization for the JESD204B IP core should be smaller by about 15 ALMs and 20 registers.

Device Family	Data Path	Number of Lanes (L)	ALMs	ALUTs	Logic Registers	Memory Block (M10K/M20K) (12) (13)
Intel Stratix 10	RX	1	873.1	1225	1307	1
		2	1344.4	1838	2018	2
		4	2293.2	3057	3474	4
		8	4412.4	5921	6576	8
	TX	1	556.5	707	890	0
		2	757.4	1059	1099	0
		4	1041.9	1532	1541	0

continued...

(12) M10K for Arria V, Cyclone V devices, M20K for Arria V GZ, Stratix V, Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

(13) The Intel Quartus Prime software may auto-fit to use MLAB when the memory size is too small. Conversion from MLAB to M20K or M10K was performed for the numbers listed above.

Device Family	Data Path	Number of Lanes (L)	ALMs	ALUTs	Logic Registers	Memory Block (M10K/M20K) (12) (13)
		8	1469.9	1950	2479	0
Intel Arria 10	RX	1	1034.8	1488	1215	0
		2	1553	2216	1839	0
		4	2688	3767	3089	0
		8	5293.5	7132	5579	0
	TX	1	726	1116	948	0
		2	895	1376	1068	0
		4	1248	1893	1308	0
		8	1923.5	2971	1787	0
Intel Cyclone 10 GX	RX	1	1011.5	1488	1207	1
		2	1512	3767	1821	2
		4	2613.5	3767	3054	4
		8	5142	7132	5515	8
	TX	1	716.5	1116	948	0
		2	892	1376	1067	0
		4	1243	1893	1308	0
		8	1925	2971	1788	0
Stratix V	RX	1	1047.2	1530	1226	0
		2	1608.7	2322	1871	0
		4	2897.2	4037	3164	0
		8	5412.5	7506	5743	0
	TX	1	711	1152	948	0
		2	926.7	1491	1086	0
		4	1345.7	2134	1359	0
		8	2114.7	3358	1907	0
Arria V	RX	1	1024.5	1516	1208	1
		2	1555.5	2302	1841	2
		4	2769.5	3951	3099	4
		8	5189	7399	5620	8
	TX	1	711.7	1149	948	0

continued...

(12) M10K for Arria V, Cyclone V devices, M20K for Arria V GZ, Stratix V, Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

(13) The Intel Quartus Prime software may auto-fit to use MLAB when the memory size is too small. Conversion from MLAB to M20K or M10K was performed for the numbers listed above.

Device Family	Data Path	Number of Lanes (L)	ALMs	ALUTs	Logic Registers	Memory Block (M10K/M20K) (12) (13)
		2	860.5	1418	1065	0
		4	1188.7	1932	1300	0
		8	1721	2854	1768	0
Arria V GZ	RX	1	1048.7	1530	1228	0
		2	1601.5	2322	1871	0
		4	2894	4037	3162	0
		8	5400.5	7506	5745	0
	TX	1	712.2	1152	948	0
		2	926.5	1491	1087	0
		4	1349.2	2134	1359	0
		8	2104.7	3358	1907	0
Cyclone V	RX	1	1022	1516	1210	1
		2	1555.5	2302	1841	2
		4	2777.5	3951	3099	4
		8	5195	7399	5622	8
	TX	1	713.5	1149	949	0
		2	867	1418	1065	0
		4	1198	1932	1301	0
		8	1709.2	2838	1768	0

Related Information

JESD204B IP Core Parameters on page 35

(12) M10K for Arria V, Cyclone V devices, M20K for Arria V GZ, Stratix V, Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

(13) The Intel Quartus Prime software may auto-fit to use MLAB when the memory size is too small. Conversion from MLAB to M20K or M10K was performed for the numbers listed above.

3. Getting Started

Related Information

- Intel FPGA Software Installation & Licensing
- Introduction to Intel FPGA IP Cores
  Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- Creating Version-Independent IP and Platform Designer Simulation Scripts
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- Project Management Best Practices
  Guidelines for efficient management and portability of your project and IP files.

3.1. Introduction to Intel FPGA IP Cores

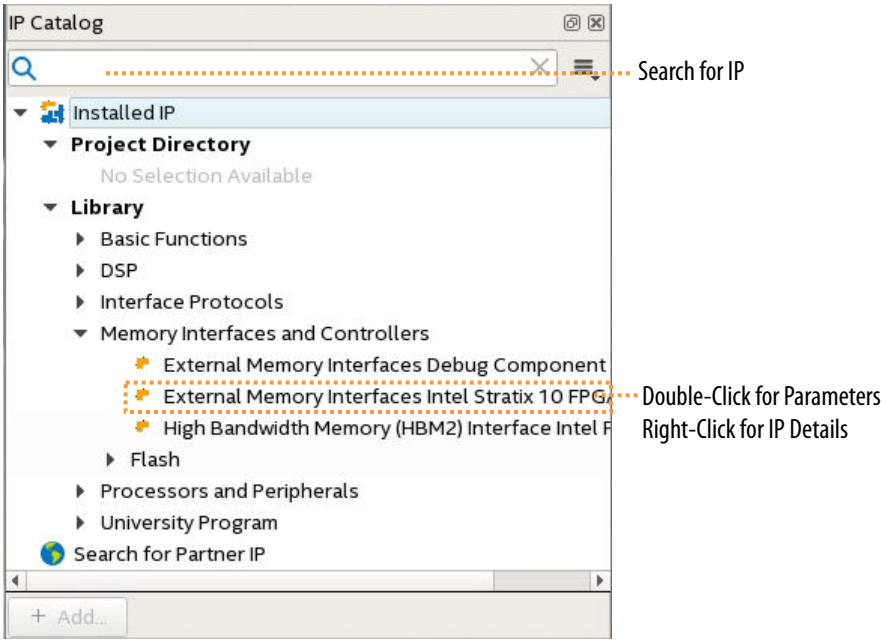
Intel and strategic IP partners offer a broad portfolio of configurable IP cores optimized for Intel FPGA devices.

The Intel Quartus Prime software installation includes the Intel FPGA IP library. Integrate optimized and verified Intel FPGA IP cores into your design to shorten design cycles and maximize performance. The Intel Quartus Prime software also supports integration of IP cores from other sources. Use the IP Catalog (**Tools ➤ IP Catalog**) to efficiently parameterize and generate synthesis and simulation files for your custom IP variation. The Intel FPGA IP library includes the following types of IP cores:

- Basic functions
- DSP functions
- Interface protocols
- Low power functions
- Memory interfaces and controllers
- Processors and peripherals

This document provides basic information about parameterizing, generating, upgrading, and simulating stand-alone IP cores in the Intel Quartus Prime software.

Figure 2. IP Catalog



3.2. 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:

Figure 3. IP Core Installation Path

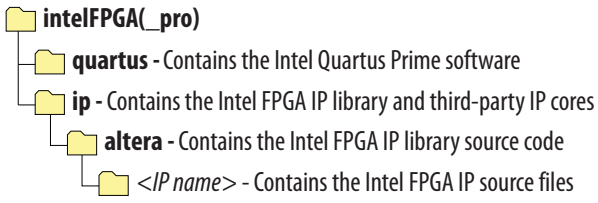


Table 9. IP Core Installation Locations

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

3.3. 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.

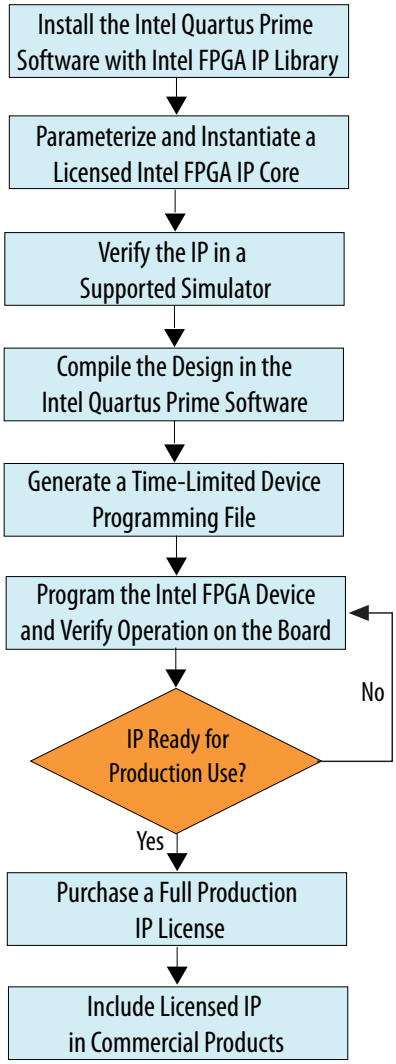
Intel FPGA IP Evaluation Mode supports the following operation modes:

- **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 4. 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 Self-Service Licensing Center or contact your local Intel FPGA representative.

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.

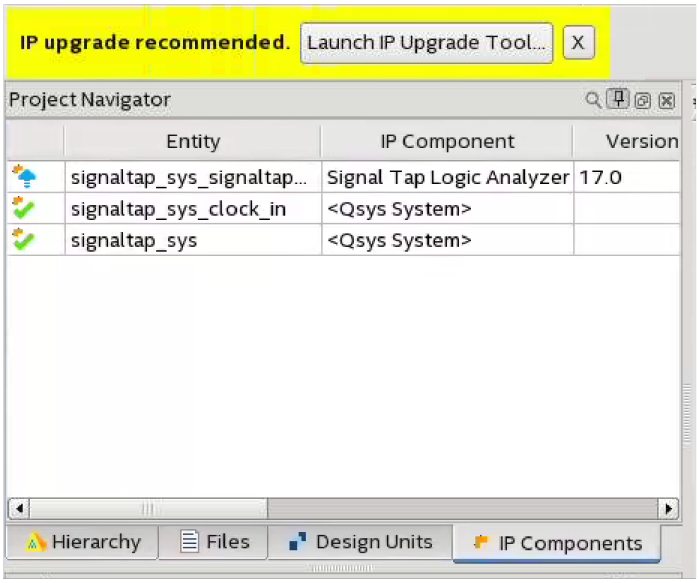
Related Information

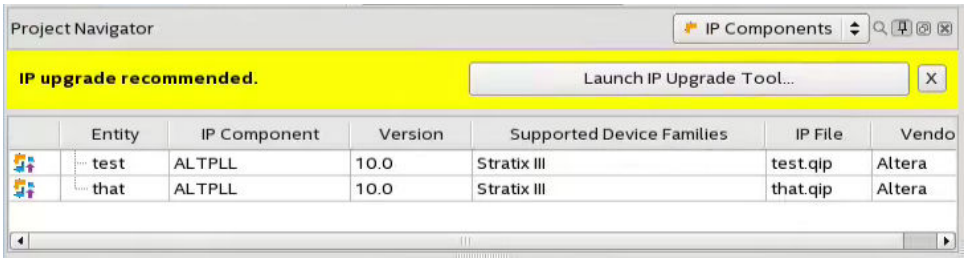
- Intel Quartus Prime Licensing Site
- Intel FPGA Software Installation and Licensing

3.4. Upgrading IP Cores

Any Intel FPGA IP variations that you generate from a previous version or different edition of the Intel Quartus Prime software, may require upgrade before compilation in the current software edition or version. The Project Navigator displays a banner indicating the IP upgrade status. Click **Launch IP Upgrade Tool** or **Project ➤ Upgrade IP Components** to upgrade outdated IP cores.

Figure 5. IP Upgrade Alert in Project Navigator





Icons in the **Upgrade IP Components** dialog box indicate when IP upgrade is required, optional, or unsupported for an IP variation in the project. Upgrade IP variations that require upgrade before compilation in the current version of the Intel Quartus Prime software.

Note: Upgrading IP cores may append a unique identifier to the original IP core entity names, without similarly modifying the IP instance name. There is no requirement to update these entity references in any supporting Intel Quartus Prime file, such as the Intel Quartus Prime Settings File (`.qsf`), Synopsys* Design Constraints File (`.sdc`), or Signal Tap File (`.stp`), if these files contain instance names. The Intel Quartus Prime software reads only the instance name and ignores the entity name in paths that specify both names. Use only instance names in assignments.

Table 10. IP Core Upgrade Status

IP Core Status	Description
IP Upgraded 	Indicates that your IP variation uses the latest version of the Intel FPGA IP core.
IP Component Outdated 	Indicates that your IP variation uses an outdated version of the IP core.
IP Upgrade Optional 	Indicates that upgrade is optional for this IP variation in the current version of the Intel Quartus Prime software. You can upgrade this IP variation to take advantage of the latest development of this IP core. Alternatively, you can retain previous IP core characteristics by declining to upgrade. Refer to the Description for details about IP core version differences. If you do not upgrade the IP, the IP variation synthesis and simulation files are unchanged and you cannot modify parameters until upgrading.
IP Upgrade Required 	Indicates that you must upgrade the IP variation before compiling in the current version of the Intel Quartus Prime software. Refer to the Description for details about IP core version differences.
IP Upgrade Unsupported 	Indicates that upgrade of the IP variation is not supported in the current version of the Intel Quartus Prime software due to incompatibility with the current version of the Intel Quartus Prime software. The Intel Quartus Prime software prompts you to replace the unsupported IP core with a supported equivalent IP core from the IP Catalog. Refer to the Description for details about IP core version differences and links to Release Notes.
IP End of Life 	Indicates that Intel designates the IP core as end-of-life status. You may or may not be able to edit the IP core in the parameter editor. Support for this IP core discontinues in future releases of the Intel Quartus Prime software.
IP Upgrade Mismatch Warning 	Provides warning of non-critical IP core differences in migrating IP to another device family.
IP has incompatible subcores 	Indicates that the current version of the Intel Quartus Prime software does not support compilation of your IP variation, because the IP has incompatible subcores

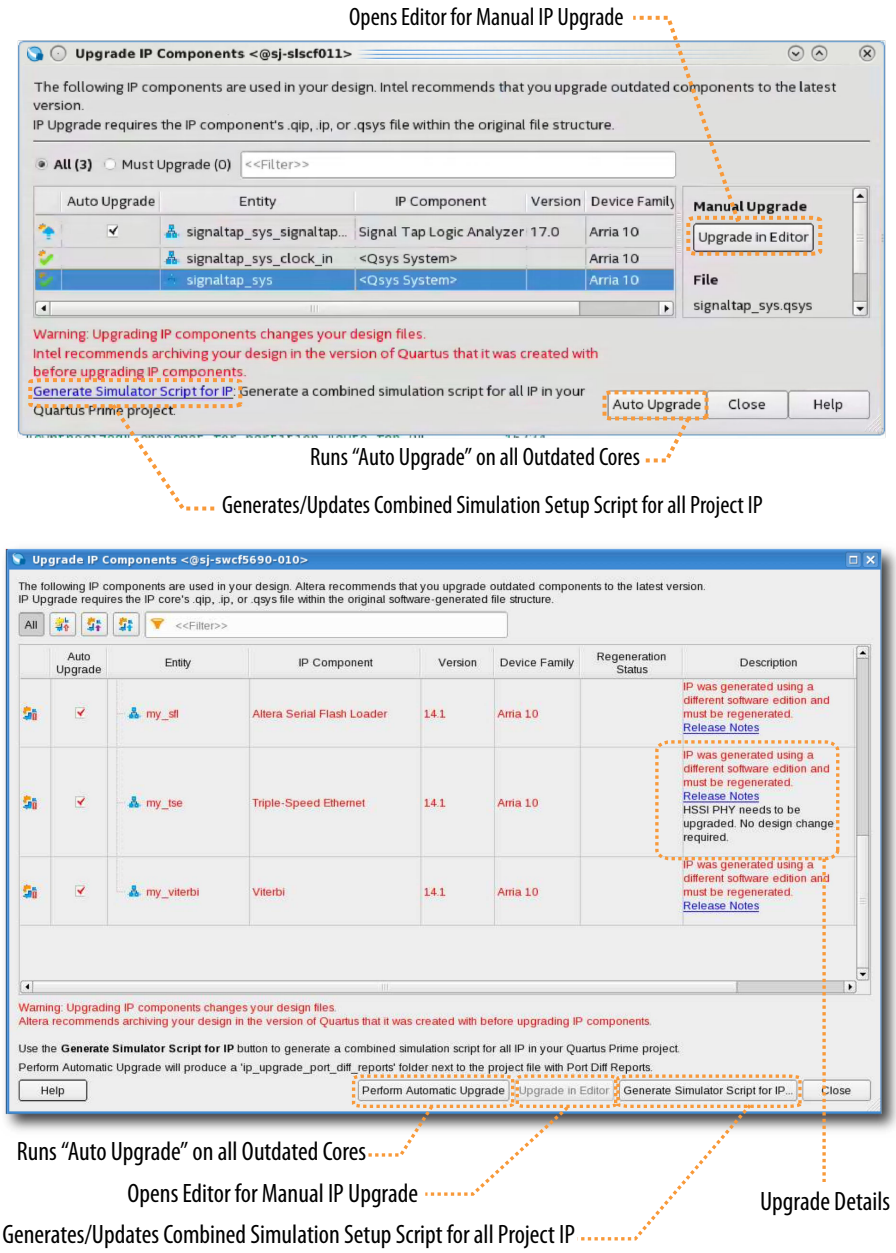
continued...

IP Core Status	Description
Compilation of IP Not Supported 	Indicates that the current version of the Intel Quartus Prime software does not support compilation of your IP variation. This can occur if another edition of the Intel Quartus Prime software, such as the Intel Quartus Prime Standard Edition, generated this IP. Replace this IP component with a compatible component in the current edition.

Follow these steps to upgrade IP cores:

1. In the latest version of the Intel Quartus Prime software, open the Intel Quartus Prime project containing an outdated IP core variation. The **Upgrade IP Components** dialog box automatically displays the status of IP cores in your project, along with instructions for upgrading each core. To access this dialog box manually, click **Project ➤ Upgrade IP Components**.
2. To upgrade one or more IP cores that support automatic upgrade, ensure that you turn on the **Auto Upgrade** option for the IP cores, and click **Auto Upgrade**. The **Status** and **Version** columns update when upgrade is complete. Example designs that any Intel FPGA IP core provides regenerate automatically whenever you upgrade an IP core.
3. To manually upgrade an individual IP core, select the IP core and click **Upgrade in Editor** (or simply double-click the IP core name). The parameter editor opens, allowing you to adjust parameters and regenerate the latest version of the IP core.

Figure 6. Upgrading IP Cores



Note: Intel FPGA IP cores older than Intel Quartus Prime software version 12.0 do not support upgrade. Intel verifies that the current version of the Intel Quartus Prime software compiles the previous two versions of each IP core. The *Intel FPGA IP Core Release Notes* reports any verification exceptions for Intel FPGA IP cores. Intel does not verify compilation for IP cores older than the previous two releases.

Related Information

Intel FPGA IP Core Release Notes

3.5. 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 prompts you to specify an IP variation name, optional ports, and output file generation options. The parameter editor generates a top-level Intel Quartus Prime IP file (`.ip`) for an IP variation in Intel Quartus Prime Pro Edition projects.

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.

Figure 7. IP Parameter Editor (Intel Quartus Prime Pro Edition)

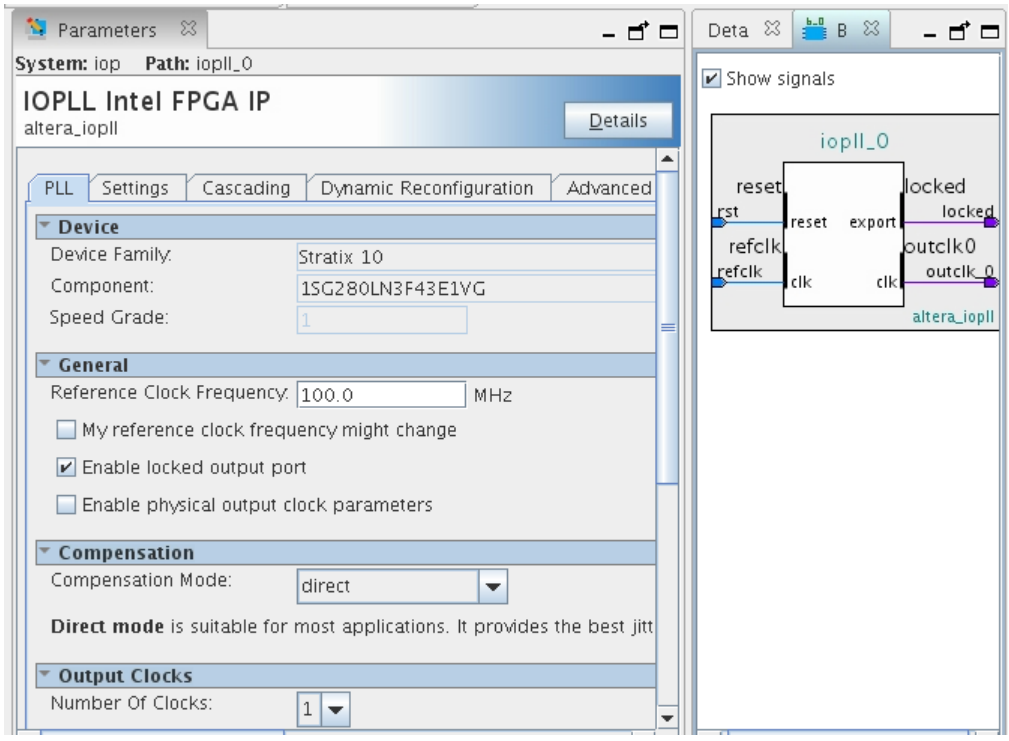
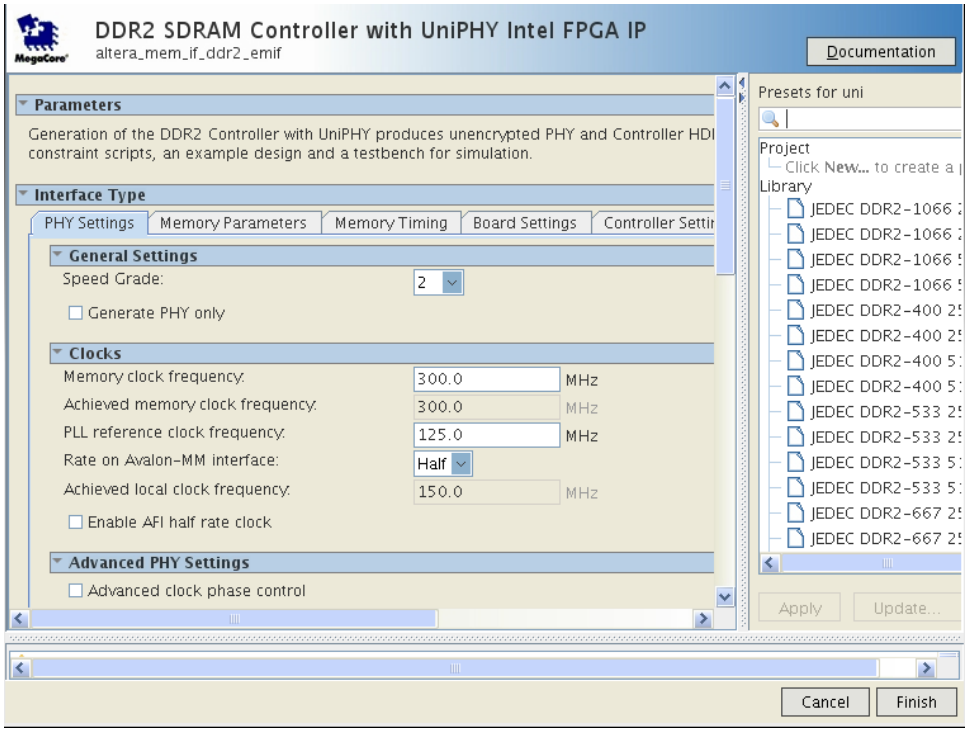


Figure 8. IP Parameter Editor (Intel Quartus Prime Standard Edition)



3.6. Design Walkthrough

This walkthrough explains how to create a JESD204B IP core design using Platform Designer in the Intel Quartus Prime software. After you generate a custom variation of the JESD204B IP core, you can incorporate it into your overall project.

3.6.1. Creating a New Intel Quartus Prime Project

You can create a new Intel Quartus Prime project with the **New Project Wizard**. This process allows you to:

- specify the working directory for the project.
- assign the project name.
- designate the name of the top-level design entity.

1. Launch the Intel Quartus Prime software.
2. On the **File** menu, click **New Project Wizard**.
3. In the **New Project Wizard: Directory, Name, Top-Level Entity** page, specify the working directory, project name, and top-level design entity name. Click **Next**.
4. In the **New Project Wizard: Add Files** page, select the existing design files (if any) you want to include in the project.(14) Click **Next**.

(14) To include existing files, you must specify the directory path to where you installed the JESD204B IP core. You must also add the user libraries if you installed the IP core Library in a different directory from where you installed the Intel Quartus Prime software.

5. In the **New Project Wizard: Family & Device Settings** page, select the device family and specific device you want to target for compilation. Click **Next**.
6. In the **EDA Tool Settings** page, select the EDA tools you want to use with the Intel Quartus Prime software to develop your project.
7. Review the summary of your chosen settings in the **New Project Wizard** window, then click **Finish** to complete the Intel Quartus Prime project creation.

3.6.2. Parameterizing and Generating the IP Core

Refer to Table 13 on page 35 for the IP core parameter values and description.

1. In the IP Catalog (**Tools ➤ IP Catalog**), locate and double-click the JESD204B Intel FPGA IP.
2. Specify a top-level name for your custom IP variation. This name identifies the IP core variation files in your project. If prompted, also specify the target Intel FPGA device family and output file HDL preference. Click **OK**.
3. In the Main tab, set the following options:
   - **Jesd204b wrapper**
   - **Data path**
   - **Jesd204b subclass**
   - **Data Rate**
   - **PCS Option**
   - **PLL Type**
   - **Bonding Mode**
   - **PLL/CDR Reference Clock Frequency**
   - **Enable Bit reversal and Byte reversal**
   - **Enable Transceiver Dynamic Reconfiguration**
   - **Enable Altera Debug Master Endpoint**
   - **Enable Capability Registers**
   - **Set user-defined IP identifier**
   - **Enable Control and Status Registers**
   - **Enable Prbs Soft Accumulators**
4. In the Jesd204b Configurations tab, select the following configurations:
   - **Common configurations (L, M, Enable manual F configuration, F, N, N', S, K)**
   - **Advanced configurations (SCR, CS, CF, HD, ECC_EN, PHADJ, ADJCNT, ADJDIR)**
5. In the Configurations and Status Registers tab, set the following configurations:
   - **Device ID**
   - **Bank ID**
   - **Lane ID**
   - **Lane checksum**

6. After parameterizing the core, go to the Example Design tab and click **Generate Example Design** to create the simulation testbench. Skip to 8 if you do not want to generate the design example.
7. Set a name for your <*example_design_directory*> and click **OK** to generate supporting files and scripts.
   The testbench and scripts are located in the <*example_design_directory*>/`ip_sim` folder.

   The **Generate Example Design** option generates supporting files for the following entities:

   - IP core for simulation—refer to Generating and Simulating the IP Core Testbench on page 41
   - IP core design example for simulation—refer to *Generating and Simulating the Design Example* section in the respective design example user guides.
   - IP core design example for synthesis—refer to Compiling the *JESD204B IP Core Design Example* section in the respective design example user guides.
8. Click **Finish** or **Generate HDL** to generate synthesis and other optional files matching your IP variation specifications. The parameter editor generates the top-level `.ip`, `.qip` or `.qsys` IP variation file and HDL files for synthesis and simulation.

   The top-level IP variation is added to the current Intel Quartus Prime project. Click **Project ➤ Add/Remove Files in Project** to manually add a `.qip` or `.qsys` file to a project. Make appropriate pin assignments to connect ports.

Note: Some parameter options are grayed out if they are not supported in a selected configuration or it is a derived parameter.

Related Information

- Design Examples for JESD204B IP Core User Guide
  Provides information about design examples for Arria V, Cyclone V, Stratix V, and Intel Arria 10 devices using Intel Quartus Prime Standard Edition software.
- JESD204B Intel Arria 10 FPGA IP Design Example User Guide
- JESD204B Intel Stratix 10 FPGA IP Design Example User Guide
- JESD204B Intel Cyclone 10 GX FPGA IP Design Example User Guide

3.6.3. Compiling the JESD204B IP Core Design

Refer to the JESD204B IP Core Design Considerations on page 30 before compiling the JESD204B IP core design.

To compile your design, click **Start Compilation** on the Processing menu in the Intel Quartus Prime software. You can use the generated `.ip` or `.qip` file to include relevant files into your project.

Related Information

- JESD204B IP Core Design Considerations on page 30
- Intel Quartus Prime Help
  More information about compilation in Intel Quartus Prime software.

3.6.4. Programming an FPGA Device

After successfully compiling your design, program the targeted Intel device with the Intel Quartus Prime Programmer and verify the design in hardware. For instructions on programming the FPGA device, refer to the *Device Programming* section in the Intel Quartus Prime Handbook.

Related Information

Device Programming

3.7. JESD204B Design Examples

The JESD204B IP core offers design examples that you can generate through the IP catalog in the Intel Quartus Prime software.

For detailed information about the JESD204B design examples, refer to following user guides:

Related Information

- Design Examples for JESD204B IP Core User Guide
  Provides information about design examples for Arria V, Cyclone V, Stratix V, and Intel Arria 10 devices using Intel Quartus Prime Standard Edition software.
- JESD204B Intel Arria 10 FPGA IP Design Example User Guide
- JESD204B Intel Stratix 10 FPGA IP Design Example User Guide
- JESD204B Intel Cyclone 10 GX FPGA IP Design Example User Guide

3.8. JESD204B IP Core Design Considerations

You must be aware of the following conditions when integrating the JESD204B IP core in your design:

- Integrating the IP core in Platform Designer
- Pin assignments
- Adding external transceiver PLL
- Timing constraints for the input clock

3.8.1. Integrating the JESD204B IP Core in Platform Designer

You can integrate the JESD204B IP core with other Platform Designer components within Platform Designer.

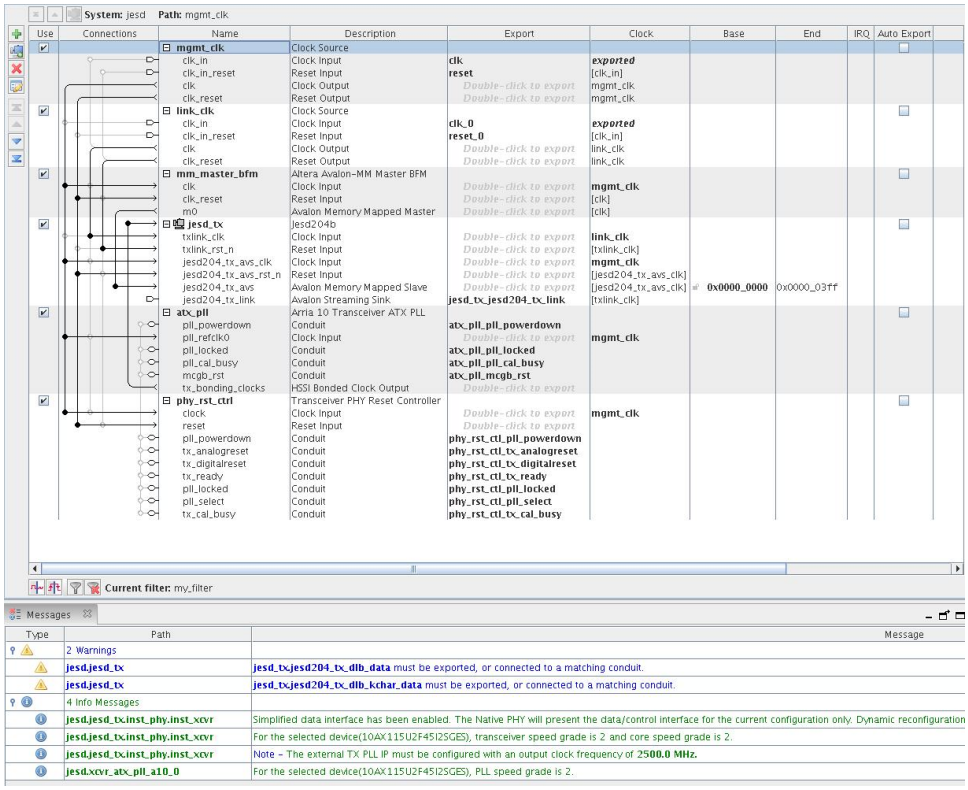
You can connect standard interfaces like clock, reset, Avalon-MM, Avalon-ST, HSSI bonded clock, HSSI serial clock, and interrupt interfaces within Platform Designer. However, for conduit interfaces, you are advised to export all those interfaces and handle them outside of Platform Designer. (15) This is because conduit interfaces are not part of the standard interfaces. Thus, there is no guarantee on compatibility between different conduit interfaces.

(15) You can also connect conduit interfaces within Platform Designer but you must create adapter components to handle all the incompatibility issues like incompatible signal type and width.

Note: The Transport Layer provided in this JESD204B IP core design example is not supported in Platform Designer. Therefore, you must export all interfaces that connect to the Transport Layer (for example, *jesd204_tx_link* interface) and connect them to a transport layer outside of Platform Designer.

Figure 9. Example of Connecting JESD204B IP Core with Other Platform Designer Components in Platform Designer

Figure shows an example of how you can connect the IP core with other Platform Designer components in Platform Designer.



Related Information

- Design Examples for JESD204B IP Core User Guide
  Provides information about design examples for Arria V, Cyclone V, Stratix V, and Intel Arria 10 devices using Intel Quartus Prime Standard Edition software.
- JESD204B Intel Arria 10 FPGA IP Design Example User Guide
- JESD204B Intel Stratix 10 FPGA IP Design Example User Guide
- JESD204B Intel Cyclone 10 GX FPGA IP Design Example User Guide

3.8.2. Pin Assignments

Set the pin assignments before you compile to provide direction to the Intel Quartus Prime software Fitter tool. You must also specify the signals that should be assigned to device I/O pins.

You can create virtual pins to avoid making specific pin assignments for top-level signals. This is useful when you want to perform compilation, but are not ready to map the design to hardware. Intel recommends that you create virtual pins for all unused top-level signals to improve timing closure.

Note: Do not create virtual pins for the clock or reset signals.

3.8.3. Adding External Transceiver PLL

The JESD204B IP core variations that target an Intel Stratix 10, Intel Arria 10, and Intel Cyclone 10 GX FPGA device require external transceiver PLLs for compilation. You are recommended to configure the PLL with medium bandwidth for the PLL settings.

JESD204B IP core variations that target an Arria V, Cyclone V, or Stratix V FPGA device contain transceiver PLLs. Therefore, no external PLLs are required for compilation.

Intel recommends that you follow the PLL recommendations in the respective Transceiver PHY user guides based on the data rates.

Note: The PMA width is 20 bits for Hard PCS and 40 bits for Soft PCS.

Related Information

- V-Series Transceiver PHY User Guide
- Intel Arria 10 Transceiver PHY User Guide
- Intel Cyclone 10 GX Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide

3.8.4. Timing Constraints For Input Clocks

When you generate the JESD204B IP core variation, the Intel Quartus Prime software generates a Synopsys Design Constraints File (`.sdc`) that specifies the timing constraints for the input clocks to your IP core.

When you generate the JESD204B IP core, your design is not yet complete and the JESD204B IP core is not yet connected in the design. The final clock names and paths are not yet known. Therefore, the Intel Quartus Prime software cannot incorporate the final signal names in the `.sdc` file that it automatically generates. Instead, you must manually modify the clock signal names in this file to integrate these constraints with the timing constraints for your full design.

This section describes how to integrate the timing constraints that the Intel Quartus Prime software generates with your IP core into the timing constraints for your design.

The Intel Quartus Prime software automatically generates the `altera_jesd204.sdc` file that contains the JESD204B IP core's timing constraints.

Three clocks are created at the input clock port:

- JESD204B TX IP core:
  - `txlink_clk`
  - `reconfig_to_xcvr[0]` (for Arria V, Cyclone V, and Stratix V devices only)
  - `reconfig_clk` (for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices only)
  - `tx_avs_clk`
- JESD204B RX IP core:
  - `rxlink_clk`
  - `reconfig_to_xcvr[0]` (for Arria V, Cyclone V, and Stratix V devices only)
  - `reconfig_clk` (for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices only)
  - `rx_avs_clk`

In a functional system design, these clocks (except for `reconfig_to_xcvr[0]` clock) are typically provided by the core PLL.

In the `.sdc` file for your project, make the following command changes:

- Specify the PLL clock reference pin frequency using the `create_clock` command.
- Derive the PLL generated output clocks from the PLL Intel FPGA IP (for Arria V, Cyclone V and Stratix V) or IOPLL Intel FPGA IP (for Intel Arria 10 and Intel Cyclone 10 GX) using the `derive_pll_clocks` command.
- For Intel Stratix 10 devices, Intel FPGA IOPLL IP core has SDC file which derives the PLL clocks based on your PLL configurations. You need not add the `derive_pll_clocks` command into your top level SDC file."
- Comment out the `create_clock` commands for the `txlink_clk`, `reconfig_to_xcvr[0]` or `reconfig_clk`, and `tx_avs_clk`, `rxlink_clk`, and `rx_avs_clk` clocks in the `altera_jesd204.sdc` file.
- Identify the base and generated clock name that correlates to the `txlink_clk`, `reconfig_clk`, and `tx_avs_clk`, `rxlink_clk`, and `rx_avs_clk` clocks using the `report_clock` command.
- Describe the relationship between base and generated clocks in the design using the `set_clock_groups` command.

After you complete your design, you must modify the clock names in your `.sdc` file to the full-design clock names, taking into account both the IP core instance name in the full design, and the design hierarchy. Be careful when adding the timing exceptions based on your design, for example, when the JESD204B IP core handles asynchronous timing between the `txlink_clk`, `rxlink_clk`, `pll_ref_clk`, `tx_avs_clk`, `rx_avs_clk`, and `reconfig_clk` (for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 only) clocks.

The table below shows an example of clock names in the `altera_jesd204.sdc` and input clock names in the user design. In this example, there is a dedicated input clock for the transceiver TX PLL and CDR at the `refclk` pin. The `device_clk` is the input to the core PLL `clkin` pin. The IP core and transceiver Avalon-MM interfaces have separate external clock sources with different frequencies.

Table 11. Example A

Original clock names in altera_jesd204.sdc	User design input clock names	Frequency (MHz)	Recommended SDC timing constraint
tx_pll_ref_clk	xcvr_tx_rx_refclk	250	create_clock -name xcvr_tx_rx_refclk -period 4.0 [get_ports xcvr_tx_rx_refclk] create_clock -name device_clk -period 8.0 [get_ports device_clk] create_clock -name jesd204_avs_clk -period 10.0 [get_ports jesd204_avs_clk] create_clock -name phy_mgmt_clk -period 13.3 [get_ports phy_mgmt_clk] derive_pll_clocks set_clock_groups -asynchronous \ -group {xcvr_tx_rx_refclk \ <base and generated clock names as reported by report_clock commands> \ } \ -group {device_clk \ <base and generated clock names as reported by report_clock commands> \ } \ -group {jesd204_avs_clk} \ -group {phy_mgmt_clk \ <base and generated clock names as reported by report_clock commands> \ }
rx_pll_ref_clk			
txlink_clk	device_clk	125	
rxlink_clk			
tx_avs_clk	jesd204_avs_clk	100	
rx_avs_clk			
reconfig_clk (16)	phy_mgmt_clk	75	

However, if your design requires you to connect the `rx_avs_clk` and `reconfig_clk` to the same clock, you need to put them in the same clock group.

The table below shows an example where the `device_clk` in this design is an input into the transceiver `refclk` pin. The IP core's Avalon-MM interface shares the same clock source as the transceiver management clock.

Table 12. Example B

Original clock names in altera_jesd204.sdc	User design input clock names	Frequency (MHz)	Recommended SDC timing constraint
tx_pll_ref_clk	device_clk	125	create_clock -name device_clk -period 8.0 [get_ports device_clk] create_clock -name mgmt_clk -period 10.0 [get_ports mgmt_clk] derive_pll_clocks set_clock_groups -asynchronous \ -group {device_clk \ <base and generated clock names as reported by report_clock commands> \ } \ -group {mgmt_clk \
rx_pll_ref_clk			
txlink_clk			
rxlink_clk			
tx_avs_clk	mgmt_clk	100	
rx_avs_clk			
reconfig_clk (17)			

continued...

(16) For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 only.

(17) For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 only.

Original clock names in altera_jesd204.sdc	User design input clock names	Frequency (MHz)	Recommended SDC timing constraint
			<base and generated clock names as reported by report_clock commands> \ }

3.9. JESD204B IP Core Parameters

Table 13. JESD204B IP Core Parameters

Parameter	Value	Description
Main Tab		
Device Family	• Arria V • Arria V GZ • Intel Arria 10 • Cyclone V • Intel Cyclone 10 GX • Stratix V • Intel Stratix 10	The targeted device family.
JESD204B Wrapper	• Base Only • PHY Only • Both Base and PHY	Select the JESD204B wrapper. • Base Only—generates the DLL only. • PHY Only—generates the transceiver PHY layer only (soft and hard PCS). • Both Base and PHY—generates both DLL and transceiver PHY layers.
Data Path	• Receiver • Transmitter • Duplex	Select the operation modes. This selection enables or disables the receiver and transmitter supporting logic. • RX—instantiates the receiver to interface to the ADC. • TX—instantiates the transmitter to interface to the DAC. • Duplex—instantiates the receiver and transmitter to interface to both the ADC and DAC.
JESD204B Subclass	• 0 • 1 • 2	Select the JESD204B subclass modes. • 0—Set subclass 0 • 1—Set subclass 1 • 2—Set subclass 2
Data Rate	1.0–16.0	Set the data rate for each lane. • Cyclone V—1.0 Gbps to 5.0 Gbps • Intel Cyclone 10 GX—2.0 Gbps to 6.25 Gbps • Arria V—1.0 Gbps to 7.5 Gbps • Arria V GZ—2.0 Gbps to 9.9 Gbps • Intel Arria 10—2.0 Gbps to 15.0 Gbps • Stratix V—2.0 Gbps to 12.5 Gbps • Intel Stratix 10—2.0 Gbps to 16.0 Gbps *Note:* The maximum data rate is limited due to different device speed grades, transceiver PMA speed grades, and PCS options. Refer to Performance and Resource Utilization on page 13 for the maximum data rate support.

continued...

Parameter	Value	Description
PCS Option	• Enabled Hard PCS • Enabled Soft PCS • Enabled PMA Direct	Select the PCS modes. • Enabled Hard PCS—use Hard PCS components. Select this option to minimize resource utilization with data rate that supports up to the limitation of the Hard PCS. *Note:* For this setting, use 8G PCS mode with 20 bits PMA width and 32-bit PCS width. • Enabled Soft PCS—use Soft PCS components. Select this option to allow higher supported data rate but increases the resource utilization. This option is applicable for all devices except Cyclone V and Arria V GT/ST. *Note:* For this setting, use 10G PCS mode with 40 bits PMA width and 40 bits PCS width. • Enabled PMA Direct—Native PHY is set to PMA Direct mode. Select this option to allow the highest supported data rate and to maximize the resource utilization. This option is applicable only for Arria V GT/ST devices. *Note:* For this setting, use PMA Direct mode with 80-bit PMA width.
PLL Type	• CMU • ATX	Select the Phase-Locked Loop (PLL) types, depending on the FPGA device family. This parameter is not applicable to Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices. • Cyclone V—CMU • Arria V—CMU • Stratix V—CMU, ATX
Bonding Mode	• Bonded • Non-bonded	Select the bonding modes. • Bonded—select this option to minimize inter-lanes skew for the transmitter datapath. • Non-bonded—select this option to disable inter-lanes skew control for the transmitter datapath. *Note:* For Stratix V, Arria V, and Cyclone V devices, the bonding type is automatically selected based on the device family and number of lanes that you set.
PLL/CDR Reference Clock Frequency	*Variable*	Set the transceiver reference clock frequency for PLL or CDR. • For Stratix V, Arria V, and Cyclone V devices, the frequency range available for you to choose depends on the PLL type and data rate that you select. • For Intel Stratix 10, Intel Cyclone 10 GX, and Intel Arria 10 devices, the frequency range available for you to choose depends on the data rate.
Enable Bit reversal and Byte reversal	On, Off	Turn on this option to set the data transmission order in MSB-first serialization. If this option is off, the data transmission order is in LSB-first serialization.
Enable Transceiver Dynamic Reconfiguration	On, Off	Turn on this option to enable dynamic data rate change. For V series devices, when you enable this option, you need to connect the reconfiguration interface to the transceiver reconfiguration controller. (18) For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, turn on this option to enable the Transceiver Native PHY reconfiguration interface.
Enable Altera Debug Master Endpoint (19)	On, Off	Turn on this option for the Transceiver Native PHY IP core to include an embedded Altera Debug Master Endpoint (ADME). This ADME connects internally to the Avalon-MM slave interface of the Transceiver Native

continued...

(18) To perform dynamic reconfiguration, you have to instantiate the Transceiver Reconfiguration Controller from the IP Catalog and connect it to the JESD204B IP core through the `reconfig_to_xcvr` and `reconfig_from_xcvr` interface.

Parameter	Value	Description
		PHY and can access the reconfiguration space of the transceiver. It can perform certain test and debug functions via JTAG using System Console. This parameter is valid only for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices and when you turn on the **Enable Transceiver Dynamic Reconfiguration** parameter. *Note:* Available only in Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
Share Reconfiguration Interface (19)	On, Off	When enabled, Transceiver Native PHY presents a single Avalon-MM slave interface for dynamic reconfiguration of all channels. In this configuration the upper address bits (Intel Stratix 10: [log_2<L> +10:11]; Intel Arria 10/Intel Cyclone 10 GX: [log_2<L>+9:10]) of the reconfiguration address bus specify the selected channel. The upper address bits only exist when L>1. Address bits (Intel Stratix 10: [10:0]; Intel Arria 10/Intel Cyclone 10 GX: [9:0]) provide the register offset address within the reconfiguration space of the selected channel. L is the number of channel. When disabled, the Native PHY IP core provides an independent reconfiguration interface for each channel. For example, when a reconfiguration interface is not shared for a four-channel Native PHY IP instance, `reconfig_address[9:0]` corresponds to the reconfiguration address bus of logical channel 0, `reconfig_address[19:10]` correspond to the reconfiguration address bus of logical channel 1, `reconfig_address[29:20]` corresponds to the reconfiguration address bus of logical channel 2, and `reconfig_address[39:30]` correspond to the reconfiguration address bus of logical channel 3. For configurations using more than one channel, this option must be enabled when **Altera Debug Master Endpoint** is enabled. *Note:* Available only in Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
Provide Separate Reconfiguration Interface for Each Channel	On, Off	When enabled, transceiver dynamic reconfiguration interface presents separate clock, reset, and Avalon-MM slave interface for each channel instead of a single wide bus. This option is only available when **Share Reconfiguration Interface** is turned off. *Note:* Available in Intel Quartus Prime Pro Edition only.
Enable Capability Registers (19)	On, Off	Turn on this option to enable capability registers, which provides high level information about the transceiver channel's configuration. *Note:* Available only in Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

continued...

(19) To support the Transceiver Toolkit in your design, you must turn on this option.

Parameter	Value	Description
Set user-defined IP identifier	0–255	Set a user-defined numeric identifier that can be read from the user identifier offset when you turn on the `Enable Capability Registers` parameter. *Note:* Available only in Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
Enable Control and Status Registers (19)	On, Off	Turn on this option to enable soft registers for reading status signals and writing control signals on the PHY interface through the embedded debug. Signals include `rx_is_locktoref`, `rx_is_locktodata`, `tx_cal_busy`, `rx_cal_busy`, `rx_serial_loopback`, `set_rx_locktodata`, `set_rx_locktoref`, `tx_analogreset`, `tx_digitalreset`, `rx_analogreset`, and `rx_digitalrest`. For more information, refer to the respective *Transceiver User Guide*. *Note:* Available only in Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
Enable Prbs Soft Accumulators (19)	On, Off	Turn on this option to set the soft logic to perform PRBS bit and error accumulation when using the hard PRBS generator and checker. *Note:* Available only in Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

JESD204B Configurations Tab		
Lanes per converter device (L)	1–8	Set the number of lanes per converter device. *Note:* Refer to Performance and Resource Utilization on page 13 for the common supported range for L and the resource utilization.
Converters per device (M)	1–256	Set the number of converters per converter device.
Enable manual F configuration	On, Off	Turn on this option to set parameter F in manual mode and enable this parameter to be configurable. Otherwise, the parameter F is in derived mode. You have to enable this parameter and configure the appropriate F value if the transport layer in your design is supporting Control Word (CF) or High Density format(HD), or both. *Note:* The auto derived F value using formula F=M*S*N\'/(8*L) may not apply if parameter CF or parameter HD, or both are enabled.
Octets per frame (F)	1, 2, 4–256	The number of octets per frame derived from the formula of F= M*N'*S/(8*L).
Converter resolution (N)	1–32	Set the number of conversion bits per converter.
Transmitted bits per sample (N')	1–32	Set the number of transmitted bits per sample (JESD204 word size, which is in nibble group). *Note:* If parameter CF equals to 0 (no control word), parameter N' must be larger than or equal to sum of parameter N and parameter CS (N' ≥ N + CS). Otherwise, parameter N' must be larger than or equal to parameter N (N'≥N).
Samples per converter per frame (S)	1–32	Set the number of transmitted samples per converter per frame.
Frames per multiframe (K)	1–32	Set the number of frames per multiframe. This value is dependent on the value of F and is derived using the following constraints: • The value of K must fall within the range of 17/F <= K <= min(32, floor (1024/F)) • The value of F*K must be divisible by 4
Enable scramble (SCR)	On, Off	Turn on this option to scramble the transmitted data or descramble the receiving data.

continued...

Control Bits (CS)	0–3	Set the number of control bits per conversion sample.
Control Words (CF)	0–32	Set the number of control words per frame clock period per link.
High density user data format (HD)	On, Off	Turn on this option to set the data format. This parameter controls whether a sample may be divided over more lanes. • On: High Density format • Off: Data should not cross the lane boundary
Enable Error Code Correction (ECC_EN)	On, Off	Turn on this option to enable error code correction (ECC) for memory blocks.
Phase adjustment request (PHADJ)	On, Off	Turn on this option to specify the phase adjustment request to the DAC. • On: Request for phase adjustment • Off: No phase adjustment This parameter is valid for Subclass 2 mode only.
Adjustment resolution step count (ADJCNT)	0–15	Set the adjustment resolution for the DAC LMFC. This parameter is valid for Subclass 2 mode only.
Direction of adjustment (ADJDIR)	• Advance • Delay	Select to adjust the DAC LMFC direction. This parameter is valid for Subclass 2 mode only.

Configurations and Status Registers Tab		
Device ID	0–255	Set the device ID number.
Bank ID	0–15	Set the device bank ID number.
Lane# ID	0–31	Set the lane ID number.
Lane# checksum	0–255	Set the checksum for each lane ID.

Related Information

Performance and Resource Utilization on page 13

3.10. JESD204B IP Core Component Files

The following table describes the generated files and other files that may be in your project directory. The names and types of generated files specified may vary depending on whether you create your design with VHDL or Verilog HDL.

Table 14. Generated Files

Extension	Description
<*variation name*>.**v** or **.vhd**	IP core variation file, which defines a VHDL or Verilog HDL description of the custom IP core. Instantiate the entity defined by this file inside of your design. Include this file when compiling your design in the Intel Quartus Prime software.
<*variation name*>**.cmp**	A VHDL component declaration file for the IP core variation. Add the contents of this file to any VHDL architecture that instantiates the IP core.
<*variation name*>**.sdc**	Contains timing constraints for your IP core variation.
<*variation name*>**.qip** or **.ip**	Contains Intel Quartus Prime project information for your IP core variation.

continued...

Extension	Description
<*variation name*>**.tcl**	Tcl script file to run in Intel Quartus Prime software.
<*variation name*>**.sip**	Contains IP core library mapping information required by the Intel Quartus Prime software.The Intel Quartus Prime software generates a **. sip** file during generation of some Intel FPGA IP cores. You must add any generated **.sip** file to your project for use by NativeLink simulation and the Intel Quartus Prime Archiver.
<*variation name*>**.spd**	Contains a list of required simulation files for your IP core.

3.11. JESD204B IP Core Testbench

The JESD204B IP core includes a testbench to demonstrate a normal link-up sequence for the JESD204B IP core with a supported configuration. The testbench also provides an example of how to control the JESD204B IP core interfaces.

The testbench instantiates the JESD204B IP core in duplex mode and connects with the Intel FPGA Transceiver PHY Reset Controller IP core. Some configurations are preset and are not programmable in the JESD204B IP core testbench. For example, the JESD204B IP core always instantiates in duplex mode even if RX or TX mode is selected in the JESD204B parameter editor.

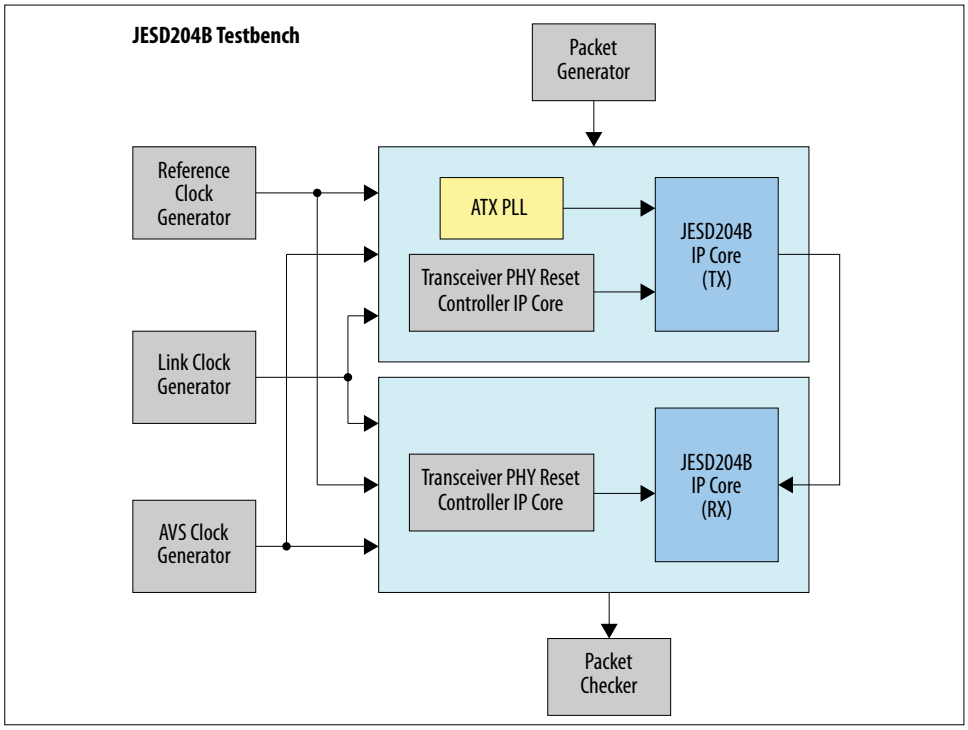
Table 15. Preset Configurations for JESD204B IP Core Testbench

Configuration	Preset Value
JESD204B Wrapper	Base and PHY (MAC and PHY)
Data Path	Simplex TX and simplex RX
PLL/CDR Reference Clock Frequency[20]	For Base only, or Simplex TX variants: • Data_rate/20 (if you turn on **Enabled Hard PCS**) • Data_rate/40 (if you turn on **Enabled Soft PCS**) • Data_rate/80 (if you turn on **Enabled PMA Direct**)
Link Clock	• Data_rate/40
AVS Clock	100 MHz

(20) For the ATX PLL supported range of reference clock frequencies, refer to the respective device datasheet.

Figure 10. JESD204B IP Core Testbench Block Diagram

The external ATX PLL is present only in the JESD204B IP core testbench targeting an Intel Arria 10 FPGA device family.



3.11.1. Generating and Simulating the IP Core Testbench

You can simulate your JESD204B IP core variation by using the provided IP core demonstration testbench.

To use the JESD204B IP core testbench, follow these steps:

1. Generate the simulation model. Refer to Generating the Testbench Simulation Model on page 41.
2. Simulate the testbench using the simulator-specific scripts that you have generated. Refer to Simulating the IP Core Testbench on page 42.

Note: Some configurations are preset and are not programmable in the JESD204B IP core testbench. For more details, refer to JESD204B IP Core Testbench on page 40 or the `README.txt` file located in the *<example_design_directory>*`/ip_sim` folder.

3.11.1.1. Generating the Testbench Simulation Model

To generate the testbench simulation model, execute the generated script (*gen_sim_verilog.tcl* or *gen_sim_vhdl.tcl*) located in the *<example_design_directory>*`/ip_sim` folder.

To run the Tcl script using the Intel Quartus Prime software, follow these steps:

1. Launch the Intel Quartus Prime software.
2. On the View menu, click **Utility Windows ➤ Tcl Console**.
3. In the **Tcl Console**, type `cd <example_design_directory>/ip_sim` to go to the specified directory.
4. Type `source gen_sim_verilog.tcl` (Verilog) or `source gen_sim_vhdl.tcl` (VHDL) to generate the simulation files.

To run the Tcl script using the command line, follow these steps:

1. Obtain the Intel Quartus Prime software resource.
2. Type `cd <example_design_directory>/ip_sim` to go to the specified directory.
3. Type `quartus_sh -t gen_sim_verilog.tcl` (Verilog) or `quartus_sh -t gen_sim_vhdl.tcl` (VHDL) to generate the simulation files.

3.11.1.2. Simulating the IP Core Testbench

Note: VHDL is not supported in VCS simulator.

Table 16. Simulation Setup Scripts

Simulator	File Directory	Script
ModelSim* - Intel FPGA Edition/ModelSim - Intel FPGA Starter Edition	*<example_design_directory>*/ip_sim/testbench/setup_scripts/mentor	msim_setup.tcl
Synopsys VCS simulator	*<example_design_directory>*/ip_sim/testbench/setup_scripts/synopsys/vcs	vcs_setup.sh
Synopsys VCS-MX simulator	*<example_design_directory>*/ip_sim/testbench/setup_scripts/synopsys/vcsmx	vcsmx_setup.sh synopsys_sim.setup
Aldec Riviera-PRO*	*<example_design_directory>*/ip_sim/testbench/setup_scripts/aldec	rivierapro_setup.tcl
Cadence NCSim	*<example_design_directory>*/ip_sim/testbench/setup_scripts/cadence	ncsim_setup.sh
Cadence Xcelium Parallel simulator	*<example_design_directory>*/ip_sim/testbench/setup_scripts/xcelium	xcelium_setup.sh

Table 17. Simulation Run Scripts

Simulator	File Directory	Script
ModelSim - Intel FPGA Edition/ModelSim - Intel FPGA Starter Edition	*<example_design_directory>*/ip_sim/testbench/mentor	run_altera_jesd204_tb.tcl
Synopsys VCS simulator	*<example_design_directory>*/ip_sim/testbench/synopsys/vcs	run_altera_jesd204_tb.sh
Synopsys VCS-MX simulator	*<example_design_directory>*/ip_sim/testbench/synopsys/vcsmx	run_altera_jesd204_tb.sh

continued...

Simulator	File Directory	Script
Aldec Riviera-PRO	`<example_design_directory>/ip_sim/testbench/aldec`	run_altera_jesd204_tb.tcl
Cadence NCSim	`<example_design_directory>/ip_sim/testbench/cadence`	run_altera_jesd204_tb.sh
Cadence Xcelium Parallel simulator	`<example_design_directory>/ip_sim/testbench/xcelium`	run_altera_jesd204_tb.sh

To simulate the testbench design using the ModelSim - Intel FPGA Edition/ModelSim - Intel FPGA Starter Edition, follow these steps:

1. Launch the ModelSim - Intel FPGA Edition/ModelSim - Intel FPGA Starter Edition.
2. On the File menu, click **Change Directory ➤ Select** <*example_design_directory*>**/ip_sim/testbench/**<*simulator name*>.
3. On the File menu, click **Load ➤ Macro file**. Select **run_altera_jesd204_tb.tcl**. This file compiles the design and runs the simulation automatically, providing a pass/fail indication on completion.

To simulate the testbench design using the Aldec Riviera-PRO simulator, follow these steps:

1. Launch the Aldec Riviera-PRO simulator.
2. On the File menu, click **Change Directory ➤ Select** <*example_design_directory*>**/ip_sim/testbench/**<*simulator name*>.
3. On the Tool menu, click **Execute Macro**. Select **run_altera_jesd204_tb.tcl**. This file compiles the design and runs the simulation automatically, providing a pass/fail indication on completion.

To simulate the testbench design using the VCS, VCS MX (in Linux), or Cadence simulators, follow these steps:

1. Launch the Synopsys VCS or VCS-MX, or Cadence NCSim or Xcelium Parallel simulator.
2. On the File menu, click **Change Directory ➤ Select** <*example_design_directory*>**/ip_sim/testbench/**<*simulator name*>.
3. Run the **run_altera_jesd204_tb.sh** file. This file compiles the design and runs the simulation automatically, providing a pass/fail indication on completion.

Related Information

Simulating Intel FPGA Designs
More information about Intel FPGA simulation models.

3.11.2. Testbench Simulation Flow

The JESD204B testbench simulation flow:

1. At the start, the system is under reset (all the components are in reset).
2. After 100 ns, the Transceiver Reset Controller IP core power up and wait for the `tx_ready` and `rx_ready` signal from the Transceiver Reset Controller IP to assert.
3. After 500ns The reset signal of the JESD204B TX Avalon-MM interface is released (go HIGH). At the next positive edge of the `link_clk` signal, the JESD204B TX link powers up by releasing its reset signal.
4. The JESD204B TX link starts transmitting K28.5 characters.
5. The reset signal of the JESD204B RX Avalon-MM interface is released (go HIGH). At the next positive edge of the `link_clk` signal, the JESD204B RX link powers up by releasing its reset signal.
6. Once the link is out of reset, a *SYSREF* pulse is generated to reset the LMFC counter inside both the JESD204B TX and RX IP core.
7. When the `txlink_ready` signal is asserted, the packet generator starts sending packets to the TX datapath.
8. The packet checker starts comparing the packet sent from the TX datapath and received at the RX datapath after the `rxlink_valid` signal is asserted.
9. The testbench reports a pass or fail when all the packets are received and compared.

The testbench concludes by checking that all the packets have been received.

If no error is detected, the testbench issues a `TESTBENCH PASSED` message stating that the simulation was successful. If an error is detected, the testbench issues a `TESTBENCH FAILED` message to indicate that the testbench has failed.

Note: For Intel Stratix 10 devices, reset deassertion staggering of TX/RX analog and digital reset happens before the assertion of TX/RX ready. The reset staggering may incur long simulation time. You may observe the staggering of TX and RX reset through `tx_analogreset_stat`, `tx_digitalreset_stat`, `rx_analogreset_stat`, and `rx_digitalreset_stat` respectively.

4. JESD204B IP Core Functional Description

The JESD204B IP core implements a transmitter (TX) and receiver (RX) block. Each block has two layers and consists of the following components:

- Media access control (MAC)—DLL block that consists of the link layer (link state machine and character replacement), CSR, Subclass 1 and 2 deterministic latency, scrambler or descrambler, and multiframe counter.
- Physical layer (PHY)—PCS and PMA block that consists of the 8B/10B encoder, word aligner, serializer, and deserializer.

You can specify the datapath and wrapper for your design and generate them separately.

The TX and RX blocks in the DLL utilizes the Avalon-ST interface to transmit or receive data and the Avalon-MM interface to access the CSRs. The TX and RX blocks operate on 32-bit data width per channel, where the frame assembly packs the data into four octets per channel. Multiple TX and RX blocks can share the clock and reset if the link rates are the same.

Figure 11. Overview of the JESD204B IP Core Block Diagram

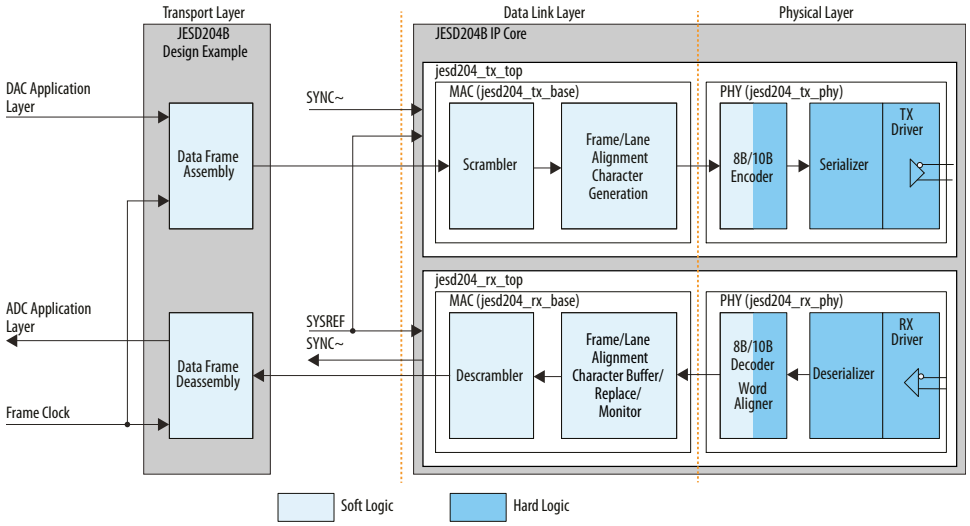
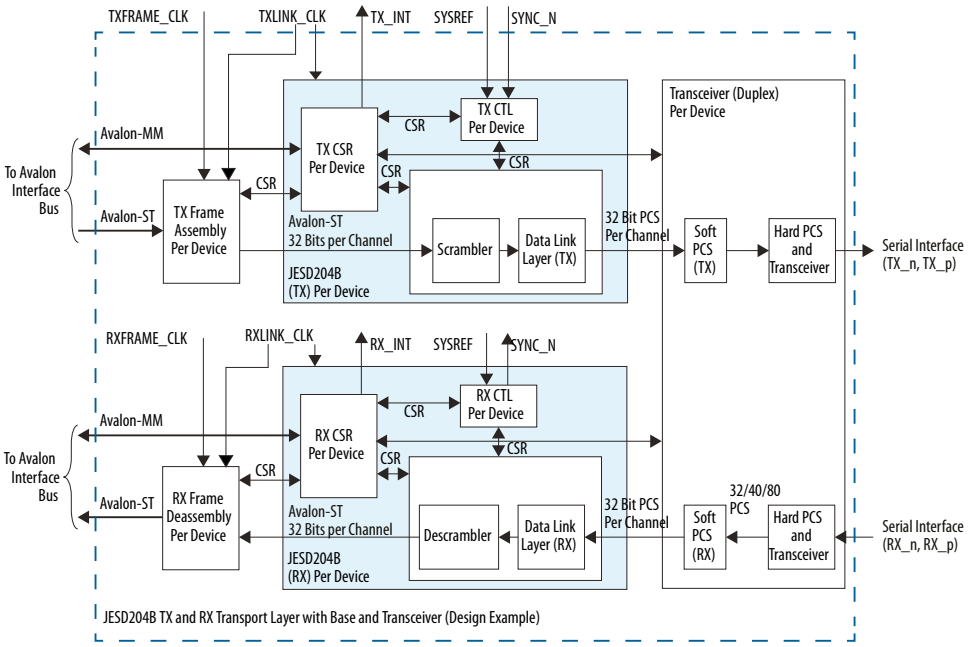


Figure 12. JESD204B IP Core TX and RX Datapath Block Diagram

The JESD204B IP core uses the Avalon-ST source and sink interfaces, with unidirectional flow of data, to transmit and receive data on the FPGA fabric interface.



32-Bits Architecture

The JESD204B IP core consist of 32-bit internal datapath per lane. This means that JESD204B IP core expects the data samples to be assembled into 32-bit data (4 octets) per lane in the transport layer before sending the data to the Avalon-ST data bus. The JESD204B IP core operates in the link clock domain. The link clock runs at (data rate/40) because it is operating in 32-bit data bus after 8B/10B encoding.

As the internal datapath of the core is 32-bits, the (F × K) value must be in the order of 4 to align the multiframe length on a 32-bit boundary. Apart from this, the deterministic latency counter values such as LMFC counter, RX Buffer Delay (RBD) counter, and Subclass 2 adjustment counter is the link clock count instead of frame clock count.

Avalon-ST Interface

The JESD204B IP core and transport layer in the design example use the Avalon-ST source and sink interfaces. There is no backpressure mechanism implemented in this core. The JESD204B IP core expects continuous stream of data samples from the upstream device.

Avalon-MM Interface

The Avalon-MM slave interface provides access to internal CSRs. The read and write data width is 32 bits (DWORD access). The Avalon-MM slave is asynchronous to the `txlink_clk`, `txframe_clk`, `rxlink_clk`, and `rxframe_clk` clock domains. You

are recommended to release the reset for the CSR configuration space first. All run-time JESD204B configurations like L, F, M, N, N', CS, CF, and HD should be set before releasing the reset for link and frame clock domain.

Each write transfer has a *writeWaitTime* of 0 cycle while a read transfer has a *readWaitTime* of 1 cycle and *readLatency* of 1 cycle.

Related Information

Avalon Interface Specification
More information about the Avalon-ST and Avalon-MM interfaces, including timing diagrams.

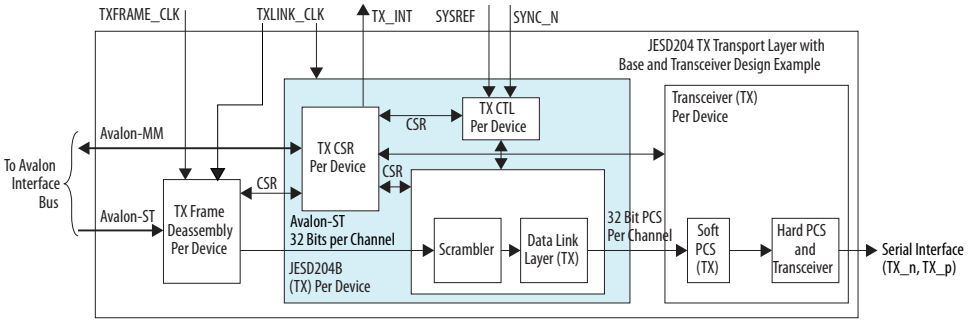
4.1. Transmitter

The transmitter block, which interfaces to DAC devices, takes one of more digital sample streams and converts them into one or more serial streams.

The transmitter performs the following functions:

- Data scrambling
- Frame or lane alignment
- Character generation
- Serial lane monitoring
- 8B/10B encoding
- Data serializer

Figure 13. Transmitter Data Path Block Diagram



The transmitter block consists of the following modules:

- TX CSR—manages the configuration and status registers.
- TX_CTL—manages the SYNC_N signal, state machine that controls the data link layer states, LMFC, and also the deterministic latency throughout the link.
- TX Scrambler and Data Link Layer—takes in 32-bits of data that implements the Initial Lane Alignment Sequence (ILAS), performs scrambling, lane insertion and frame alignment of characters.

4.1.1. TX Data Link Layer

The JESD204B IP core TX data link layer includes three phases to establish a synchronized link—Code Group Synchronization (CGS), Initial Lane Synchronization (ILAS), and User Data phase.

4.1.1.1. TX CGS

The CGS phase is achieved through the following process:

- Upon reset, the converter device (RX) issues a synchronization request by driving SYNC_N low. The JESD204B TX IP core transmits a stream of /K/ = /K28.5/ symbols. The receiver synchronizes when it receives four consecutive /K/ symbols.
- For Subclass 0, the RX converter devices deassert SYNC_N signal at the frame boundary. After all receivers have deactivated their synchronization requests, the JESD204B TX IP core continues to emit /K/ symbols until the start of the next frame. The core proceeds to transmit ILAS data sequence or encoded user data if `csr_lane_sync_en` signal is disabled.
- For Subclass 1 and 2, the RX converter devices deassert SYNC_N signal at the LMFC boundary. After all receivers deactivate the SYNC_N signal, the JESD204B TX IP core continues to transmit /K/ symbols until the next LMFC boundary. At the next LMFC boundary, the JESD204B IP core transmits ILAS data sequence. (There is no programmability to use a later LMFC boundary.)

4.1.1.2. TX ILAS

When lane alignment sequence is enabled through the `csr_lane_sync_en` register, the ILAS sequence is transmitted after the CGS phase. The ILAS phase takes up four multiframes. For Subclass 0 mode, you can program the CSR (`csr_ilas_multiframe`) to extend the ILAS phase to a maximum of 256 multiframes before transitioning to the encoded user data phase. The ILAS data is not scrambled regardless of whether scrambling is enabled or disabled.

The multiframe has the following structure:

- Each multiframe starts with a /R/ character (K28.0) and ends with a /A/ character (K28.3)
- The second multiframe transmits the ILAS configuration data. The multiframe starts with /R/ character (K28.0), followed by /Q/ character (K28.4), and then followed by the link configuration data, which consists of 14 octets as illustrated in the table below. It is then padded with dummy data and ends with /A/ character (K28.3), marking the end of multiframe.
- Dummy octets are an 8-bit counter and is always reset when it is not in ILAS phase.
- For a configuration of more than four multiframes, the multiframe follows the same rule above and is padded with dummy data in between /R/ character and /A/ character.

Table 18. Link Configuration Data Transmitted in ILAS Phase

Configuration Octet	Bits								Description
	MSB	6	5	4	3	2	1	LSB	
0	DID[7:0]								DID = Device ID
1	ADJCNT[3:0]				BID[3:0]				ADJCNT = Number of adjustment resolution steps (21) BID = Bank ID
2	0	ADJDIR	PHADJ	LID[4:0]					ADJDIR = Direction to adjust DAC LMFC (21) PHADJ = Phase adjustment request (21) LID = Lane ID
3	SCR	0	0	L[4:0]					SCR = Scrambling enabled/ disabled L = Number of lanes per device (link)
4	F[7:0]								F = Number of octets per frame per lane
5	0	0	0	K[4:0]					K = Number of frames per multiframe
6	M[7:0]								M = Number of converters per device
7	CS[1:0]		0	N[4:0]					CS = Number of control bits per sample N = Converter resolution
8	SUBCLASSV[2:0]			N_PRIME[4:0]					SUBCLASSV = Subclass version N_PRIME = Total bits per sample
9	JESDV[2:0]			S[4:0]					JESDV = JESD204 version S = Number of samples per converter per frame
10	HD	0	0	CF[4:0]					HD = High Density data format CF = Number of control words per frame clock per link
11	RES1[7:0]								RES1 = Reserved. Set to 8'h00
12	RES2[7:0]								RES2 = Reserved. Set to 8'h00
13	FCHK[7:0]								FCHK is the modulus 256 of the sum of the 13 configuration octets above. For Intel Arria 10, Intel Cyclone 10 GX, Arria V, Cyclone V, and Stratix V devices, if you change any of the octets during run time, make sure to update the new FCHK value in the register.

(21) Applies to Subclass 2 only.

The JESD204B TX IP core also supports debug feature to continuously stay in ILAS phase without exiting. You can enable this feature by setting the bit in `csr_ilas_loop` register. There are two modes of entry:

- RX asserts SYNC_N and deasserts it after CGS phase. This activity triggers the ILAS phase and the CSR stays in ILAS phase indefinitely until this setting changes.
- Link reinitialization through CSR is initiated. The JESD204B IP core transmits /K/ character and causes the RX converter to enter CGS phase. After RX deasserts SYNC_N, the CSR enters ILAS phase and stays in that phase indefinitely until this setting changes.

In ILAS loop, the multiframe transmission is the same where /R/ character (K28.0) marks the start of multiframe and /A/ character (K28.3) marks the end of multiframe, with dummy data in between. The dummy data is an increment of Dx.y.

4.1.1.3. User Data Phase

During the user data phase, character replacement at the end of frame and end of multiframe is opportunistically inserted so that there is no additional overhead for data bandwidth.

Character replacement for non-scrambled data

The character replacement for non-scrambled mode in the IP core follows these JESD204B specification rules:

- At end of frame (not coinciding with end of multiframe), which equals the last octet in the previous frame, the transmitter replaces the octet with /F/ character (K28.7). However, the original octet is encoded if an alignment character was transmitted in the previous frame.
- At the end of a multiframe, which equals to the last octet in the previous frame, the transmitter replaces the octet with /A/ character (K28.3), even if a control character was already transmitted in the previous frame.

For devices that do not support lane synchronization, only /F/ character replacement is done. At every end of frame, regardless of whether the end of multiframe equals to the last octet in previous frame, the transmitter encodes the octet as /F/ character (K28.7) if it fits the rules above.

Character replacement for scrambled data

The character replacement for scrambled data in the IP core follows these JESD204B specification rules:

- At end of frame (not coinciding with end of multiframe), which equals to 0xFC (D28.7), the transmitter encodes the octet as /F/ character (K28.7).
- At end of multiframe, which equals to 0x7C, the transmitter replaces the current last octet as /A/ character (K28.3).

For devices that do not support lane synchronization, only /F/ character replacement is done. At every end of frame, regardless of whether the end of multiframe equals to 0xFC (D28.7), the transmitter encodes the octet as /F/ character (K28.7) if it fits the rules above.

4.1.2. TX PHY Layer

The 8B/10B encoder encodes the data before transmitting them through the serial line. The 8B/10B encoding has sufficient bit transition density (3-8 transitions per 10-bit symbol) to allow clock recovery by the receiver. The control characters in this scheme allow the receiver to:

- synchronize to 10-bit boundary.
- insert special character to mark the start and end of frames and start and end of multiframes.
- detect single bit errors.

The JESD204B IP core supports transmission order from MSB first as well as LSB first. For MSB first transmission, the serialization of the left-most bit of 8B/10B code group (bit "a") is transmitted first.

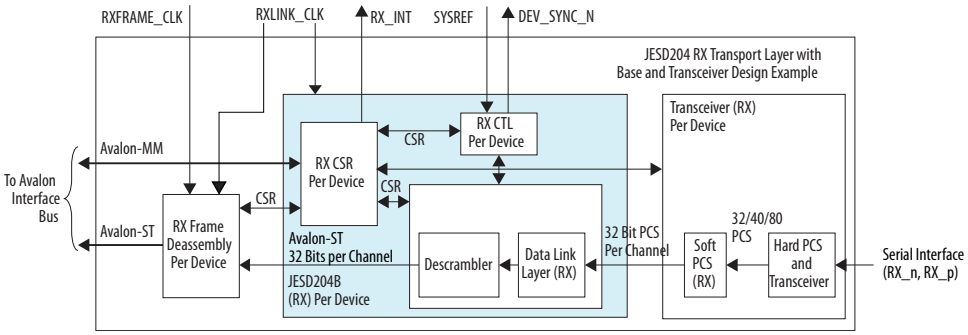
4.2. Receiver

The receiver block, which interfaces to ADC devices, receives the serial streams from one or more TX blocks and converts the streams into one or more sample streams.

The receiver performs the following functions:

- Data deserializer
- 8B/10B decoding
- Lane alignment
- Character replacement
- Data descrambling

Figure 14. Receiver Data Path Block Diagram



The receiver block includes the following modules:

- RX CSR—manages the configuration and status registers.
- RX_CTL—manages the SYNC_N signal, state machine that controls the data link layer states, LMFC, and also the buffer release, which is crucial for deterministic latency throughout the link.
- RX Scrambler and Data Link Layer—takes in 32-bits of data that decodes the ILAS, performs descrambling, character replacement as per the JESD204B specification, and error detection (code group error, frame and lane realignment error).

4.2.1. RX Data Link Layer

The JESD204B IP core RX data link layer buffers incoming user data on all lanes until the RX elastic buffers can be released. Special character substitution are done in the TX link so that the RX link can execute frame and lane alignment monitoring based on the JESD204B specification.

4.2.1.1. RX CGS

The CGS phase is the link up phase that monitors the detection of /K28.5/ character.

The CGS phase is achieved through the following process:

- Once the word boundary is aligned, the RX PHY layer detects the /K28.5/ 20-bit boundary and indicate that the character is valid.
- The receiver deasserts SYNC_N on the next frame boundary (for Subclass 0) or on the next LMFC boundary (for Subclass 1 and 2) after the reception of four successive /K/ characters.
- After correct reception of another four 8B/10B characters, the receiver assumes full code group synchronization. Error detected in this state machine is the code group error. Code group error always trigger link reinitialization through the assertion of SYNC_N signal and this cannot be disabled through the CSR. The CS state machine is defined as CS_INIT, CS_CHECK, and CS_DATA.
- The minimum duration for a synchronization request on the SYNC_N is five frames plus nine octets.

4.2.1.2. Frame Synchronization

After CGS phase, the receiver assumes that the first non-/K28.5/ character marks the start of frame and multiframe. If the transmitter emits an initial lane alignment sequence, the first non-/K28.5/ character is /K28.0/. Similar to the JESD204B TX IP core, the csr_lane_sync_en is set to 1 by default, thus the RX core detects the /K/ character to /R/ character transition. If the csr_lane_sync_en is set to 0, the RX core detects the /K/ character to the first data transition. An ILAS error and unexpected /K/ character is flagged if either one of these conditions are violated.

When csr_lane_sync_en is set to 0, you have to disable data checking for the first 16 octets of data as the character replacement block takes 16 octets to recover the end-of-frame pointer for character replacement. When csr_lane_sync_en is set to 1 (default JESD204B setting), the number of octets to be discarded depends on the scrambler or descrambler block.

The receiver assumes that a new frame starts in every F octets. The octet counter is used for frame alignment and lane alignment.

Related Information

Scrambler/Descrambler on page 58

4.2.1.3. Frame Alignment

The frame alignment is monitored through the alignment character /F/. The transmitter inserts this character at the end of frame. The /A/ character indicates the end of multiframe. The character replacement algorithm depends on whether scrambling is enabled or disabled, regardless of the `csr_lane_sync_en` register setting.

The alignment detection process:

- If two successive valid alignment characters are detected in the same position other than the assumed end of frame—without receiving a valid or invalid alignment character at the expected position between two alignment characters—the receiver realigns its frame to the new position of the received alignment characters.
- If lane realignment can result in frame alignment error, the receiver issues an error.

In the JESD204B RX IP core, the same flexible buffer is used for frame and lane alignment. Lane realignment gives a correct frame alignment because lane alignment character doubles as a frame alignment character. A frame realignment can cause an incorrect lane alignment or link latency. The course of action is for the RX to request for reinitialization through SYNC_N. (22)

4.2.1.4. Lane Alignment

After the frame synchronization phase has entered FS_DATA, the lane alignment is monitored via /A/ character (/K28.3/) at the end of multiframe. The first /A/ detection in the ILAS phase is important for the RX core to determine the minimum RX buffer release for inter-lane alignment. There are two types of error that is detected in lane alignment phase:

- Arrival of /A/ character from multiple lanes exceed one multiframe.
- Misalignment detected during user data phase.

The realignment rules for lane alignment are similar to frame alignment:

- If two successive and valid /A/ characters are detected at the same position other than the assumed end of multiframe—without receiving a valid/invalid /A/ character at the expected position between two /A/ characters—the receiver aligns the lane to the position of the newly received /A/ characters.
- If a recent frame alignment causes the loss of lane alignment, the receiver realigns the lane frame—which is already at the position of the first received /A/ character—at the unexpected position.

(22) Dynamic frame realignment and correction is not supported.

4.2.1.5. ILAS Data

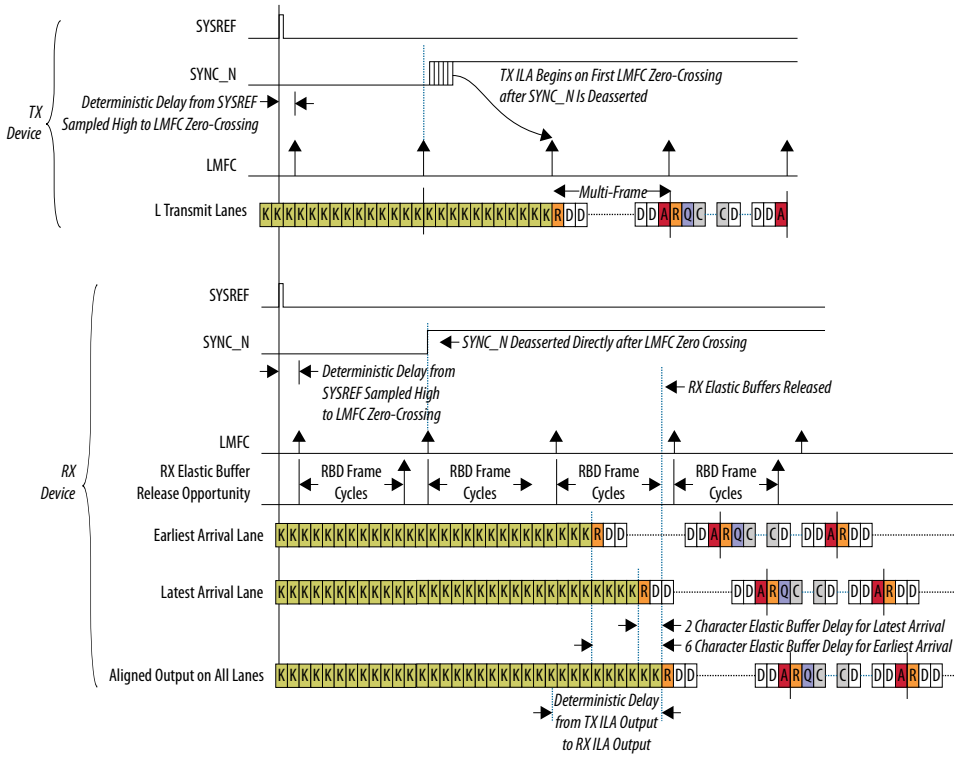
The JESD204B RX IP core captures 14 octets of link configuration data that are transmitted on the 2nd multiframe of the ILAS phase. The receiver waits for the reception of /Q/ character that marks the start of link configuration data and then latch it into ILAS octets, which are per lane basis. You can read the 14 octets captured in the link configuration data through the CSR. You need to first set the `csr_ilas_data_sel` register to select which link configuration data lane it is trying to read from. Then, proceed to read from the `csr_ilas_octet` register.

4.2.1.6. Initial Lane Synchronization

The receivers in Subclass 1 and Subclass 2 modes store data in a memory buffer (Subclass 0 mode does not store data in the buffer but immediately releases them on the frame boundary as soon as the latest lane arrives.). The RX IP core detects the start of multiframe of user data per lane and then wait for the latest lane data to arrive. The latest data is reported as RBD count (`csr_rbd_count`) value which you can read from the status register. This is the earliest release opportunity of the data from the deskew FIFO (referred to as RBD offset).

The JESD204B RX IP core supports RBD release at 0 offset and also provides programmable offset through RBD count. By default, the RBD release can be programmed through the `csr_rbd_offset` to release at the LMFC boundary. If you want to implement an early release mechanism, program it in the `csr_rbd_offset` register. The `csr_rbd_offset` and `csr_rbd_count` is a counter based on the link clock boundary (not frame clock boundary). Therefore, the RBD release opportunity is at every four octets.

Figure 15. Subclass 1 Deterministic Latency and Support for Programmable Release Opportunity



4.2.2. RX PHY Layer

The word aligner block identifies the MSB and LSB boundaries of the 10-bit character from the serial bit stream. Manual alignment is set because the /K/ character must be detected in either LSB first or MSB first mode. When the programmed word alignment pattern is detected in the current word boundary, the PCS indicates a valid pattern in the `rx_sync_status` (mapped as `pcs_valid` to the IP core). The code synchronization state is detected after the detection of the /K/ character boundary for all lanes.

In a normal operation, whenever synchronization is lost, the JESD204B RX IP core always return back to the `CS_INIT` state where the word alignment is initiated. For debug purposes, you can bypass this alignment by setting the `csr_patternalign_en` register to 0.

The 8B/10B decoder decode the data after receiving the data through the serial line. The JESD204B IP core supports transmission order from MSB first as well as LSB first.

The PHY layer can detect 8B/10B not-in-table (NIT) error and also running disparity error.

4.3. Operation

4.3.1. Operating Modes

The JESD204B IP core supports Subclass 0, 1, and 2 operating modes.

4.3.1.1. Subclass 0 Operating Mode

The JESD204B IP core maintains a LMFC counter that counts from 0 to (F × K/4)–1 and wraps around again. The LMFC counter starts counting at the deassertion of *SYNC_N* signal from multiple DACs after synchronization. This is to align the LMFC counter upon transmission and can only be done after all the converter devices have deasserted its synchronization request signal.

4.3.1.2. Subclass 1 Operating Mode

The JESD204B IP core maintains a LMFC counter that counts from 0 to (F × K/4)–1 and wraps around again. The LMFC counter resets within two link clock cycles after converter devices issue a common *SYSREF* frequency to all the transmitters and receivers. The *SYSREF* frequency must be the same for converter devices that are grouped and synchronized together.

Table 19. Example of SYSREF Frequency Calculation

In this example, you can choose to perform one of the following options:

- provide two *SYSREF* and device clock, where the ADC groups share both the device clock and *SYSREF* (18.75 MHz and 9.375 MHz)
- provide one *SYSREF* (running at 9.375 MHz) and device clock for all the ADC and DAC groups because the *SYSREF* period in the DAC is a multiplication of *n* integer.

Group	Configuration	SYSREF Frequency
ADC Group 1 (2 ADCs)	• LMF = 222 • K = 16 • Data rate = 6 Gbps	(6 GHz / 40) / (2 x 16 / 4) = 18.75 MHz
ADC Group 2 (2 ADCs)	• LMF = 811 • K = 32 • Data rate = 6 Gbps	(6 GHz / 40) / (1 x 32 / 4) = 18.75 MHz
DAC Group 3 (2 DACs)	• LMF = 222 • K = 16 • Data rate = 3 Gbps	(3 GHz / 40) / (2 x 16 / 4) = 9.375 MHz

4.3.1.3. Subclass 2 Operating Mode

The JESD204B IP core maintains a LMFC counter that counts from 0 to (F × K/4)–1 and wraps around again. The LMFC count starts upon reset and the logic device always acts as the timing master. To support Subclass 2 for multi-link device, you must deassert the resets for all JESD204B IP core links synchronously at the same clock edge. This deassertion ensures that the internal LMFC vaunter is aligner across multi-link. The converters adjust their own internal LMFC to match the master's counter. The alignment of LMFC within the system relies on the correct alignment of SYNC_N signal deassertion at the LMFC boundary.

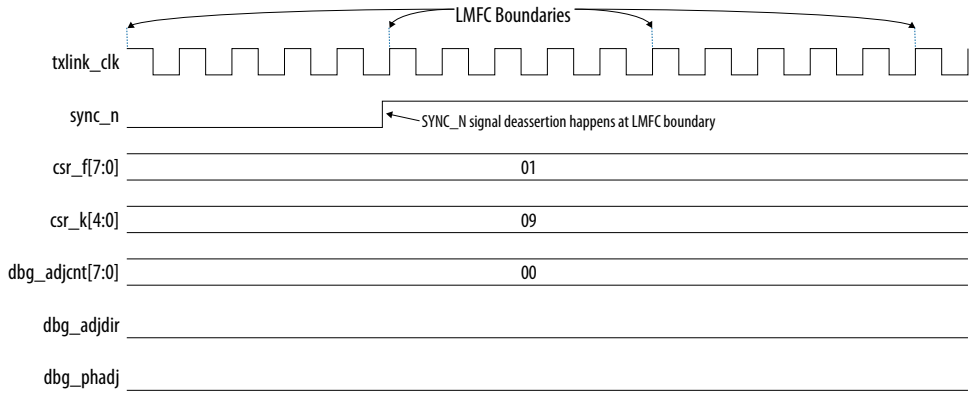
The alignment of LMFC to RX logic is handled within the TX converter. The RX logic releases SYNC_N at the LMFC tick and the TX converter adjust its internal LMFC to match the RX LMFC.

For the alignment of LMFC to the TX logic, the JESD204B TX IP core samples SYNC_N from the DAC receiver and reports the relative phase difference between the DAC and TX logic device LMFC in the TX CSR (`dbg_phadj`, `dbg_adjdir`, and `dbg_adjcnt`). Based on the reported value, you can calculate the adjustment required. Then, to initiate the link reinitialization through the CSR, set the value in the TX CSR (`csr_phadj`, `csr_adjdir`, and `csr_adjcnt`). The values on the phase adjustment are embedded in bytes 1 and 2 of the ILAS sequence that is sent to the DAC during link initialization. On the reception of the ILAS, the DAC adjusts its LMFC phase by step count value and sends back an error report with the new LMFC phase information. This process may be repeated until the LMFC at the DAC and the logic device are aligned.

Table 20. `dbg_phadj`, `dbg_adjdir` and `dbg_adjcnt` Values for Different SYNC_N Deassertions

Case	SYNC_N Signal Deassertion	`dbg_phadj` Value	`dbg_adjdir` Value	`dbg_adjcnt` Value
1	Happens at LMFC boundary[23]	0	—	—
2	Happens at LMFC count value that is equals or less than half of FxK/4 value	1	0	Number of link clock cycles from the LMFC boundary to the detection of SYNC_N signal deassertion
3	Happens at LMFC count value that is more than half of FxK/4 value	1	1	Number of link clock cycles from detection of the SYNC_N signal deassertion to the next LMFC boundary

Figure 16. Timing Diagram Example for Case 1



(23) No adjustment is required.

Figure 17. Timing Diagram Example for Case 2

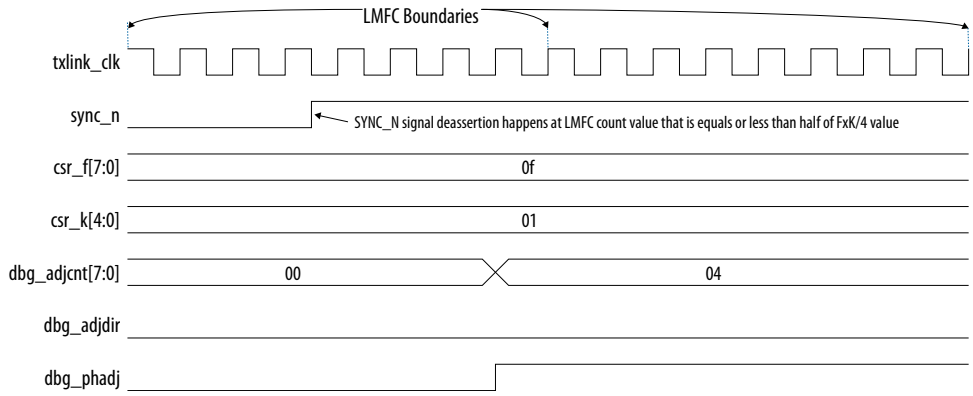
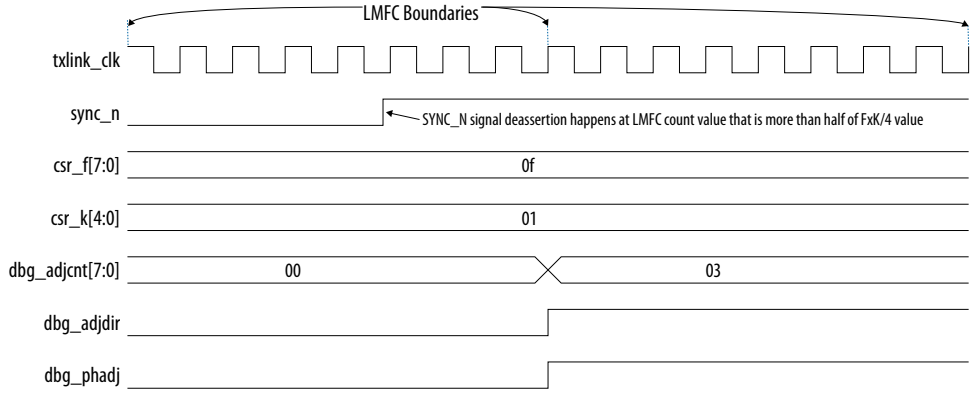


Figure 18. Timing Diagram Example for Case 3



4.3.2. Scrambler/Descrambler

Both the scrambler and descrambler are designed in a 32-bit parallel implementation and the scrambling/descrambling order starts from first octet with MSB first.

The JESD204B TX and RX IP core support scrambling by implementing a 32-bit parallel scrambler in each lane. The scrambler and descrambler are located in the JESD204B IP MAC interfacing to the Avalon-ST interface. You can enable or disable scrambling and this option applies to all lanes. Mixed mode operation, where scrambling is enabled for some lanes, is not permitted.

The scrambling polynomial:

$1 + x^{14} + x^{15}$

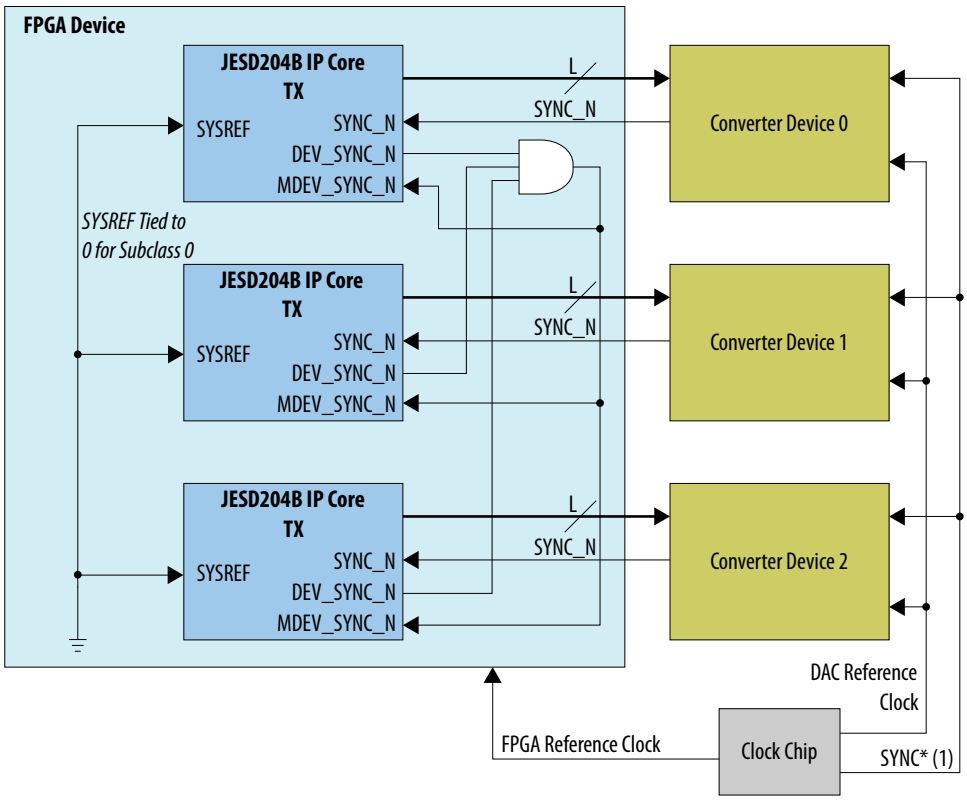
The descrambler can self-synchronize in eight octets. In a typical application where the reset value of the scrambler seed is different from the converter device to FPGA logic device, the correct user data is recovered in the receiver in two link clocks (due to the 32-bit architecture). The PRBS pattern checker on the transport layer should always disable checking of the first eight octets from the JESD204B RX IP core.

4.3.3. SYNC_N Signal

For Subclass 0 implementation, the SYNC_N signal from the DAC converters in the same group path must be combined.

In some applications, multiple converters are grouped together in the same group path to sample a signal (referred as multipoint link). The FPGA can only start the LMFC counter and its transition to ILAS after all the links deassert the synchronization request. The JESD204B TX IP core provides three signals to facilitate this application. The SYNC_N is the direct signal from the DAC converters. The error signaling from SYNC_N is filtered and sent out as dev_sync_n signal. For Subclass 0, you need to multiplex all the dev_sync_n signals in the same multipoint link and then input them to the IP core through mdev_sync_n signal.

Figure 19. Subclass 0 — Combining the SYNC_N Signal for JESD204B TX IP Core



Note:

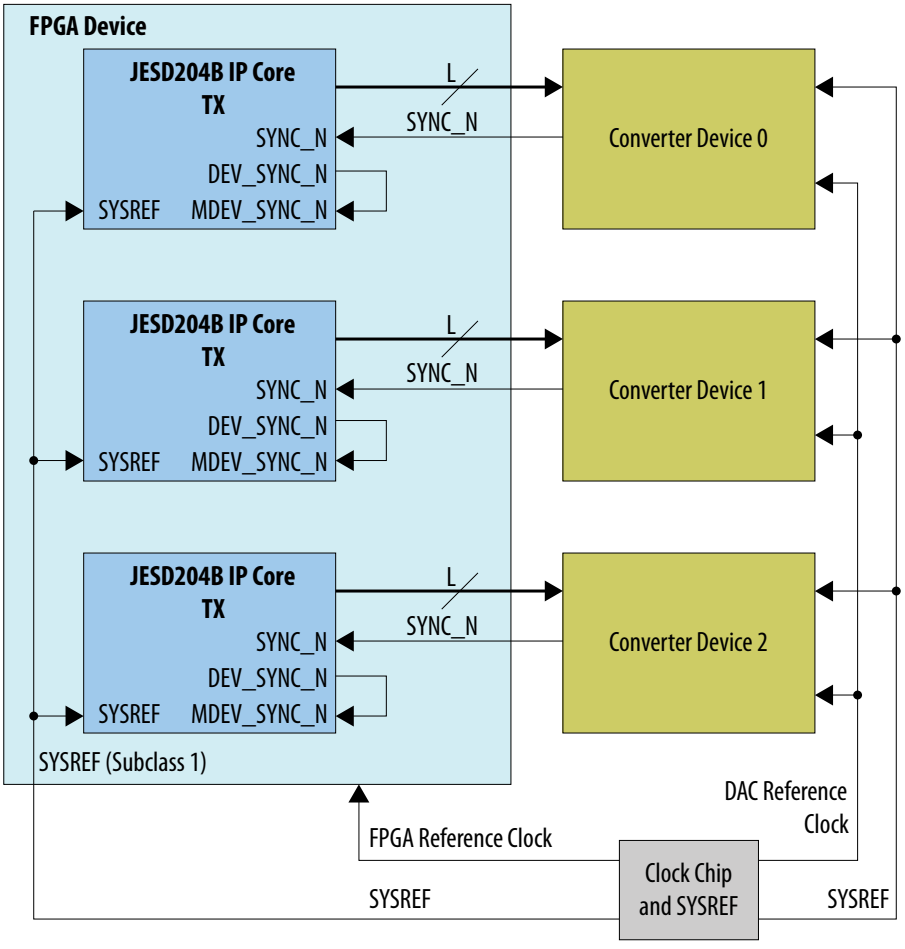
1. SYNC* is not associated to SYNC_N in the JESD204B specification. SYNC* refers to JESD204A (Subclass 0) converter devices that may support synchronization via additional SYNC signalling.

For Subclass 1 implementation, you may choose to combine or not to combine the SYNC_N signal from the converter device. If you implement two ADC converter devices as a multipoint link and one of the converter is unable to link up, the functional link still operates. You must manage the trace length for the SYSREF signal and also the differential pair to minimize skew.

The SYNC_N is the direct signal from the DAC converters. The error signaling from SYNC_N is filtered and sent out as dev_sync_n output signal. The dev_sync_n signal from the JESD204B TX IP core must loopback into the mdev_sync_n signal of the same instance without combining the SYNC_N signal.

You must set the same RBD offset value (csr_rbd_offset) to all the JESD204B RX IP cores within the same multipoint link for the RBD release (the latest lane arrival for each of the links). The JESD204B RX IP core deskews and outputs the data when the RBD offset value is met. The total latency is consistent in the system and is also the same across multiple resets. Setting a different RBD offset to each link or setting an early release does not guarantee deterministic latency and data alignment.

Figure 20. Subclass 1 — Combining the SYNC_N Signal for JESD204B TX IP Core



Related Information

Programmable RBD Offset on page 97

4.3.4. Link Reinitialization

The JESD204B TX and RX IP core support link reinitialization.

There are two modes of entry for link reinitialization:

- Hardware initiated link reinitialization:
  - For TX, the reception of SYNC_N for more than five frames and nine octets triggers link reinitialization.
  - For RX, the loss of code group synchronization, frame alignment and lane alignment errors cause the IP core to assert SYNC_N and request for link reinitialization.
- Software initiated link reinitialization—both the TX and RX IP core allow software to request for link reinitialization.
  - For TX, the IP core transmits /K/ character and wait for the receiver to assert SYNC_N to indicate that it has entered CS_INIT state.
  - For RX, the IP core asserts SYNC_N to request for link reinitialization.

Hardware initiated link reinitialization can be globally disabled through the `csr_link_reinit_disable` register for debug purposes.

Hardware initiated link reinitialization can be issued as interrupt depending on the error type and interrupt error enable. If lane misalignment has been detected as a result of a phase change in local timing reference, the software can rely on this interrupt trigger to initiates a LMFC realignment. The realignment process occurs by first resampling *SYSREF* and then issuing a link reinitialization request.

4.3.5. Link Startup Sequence

Set the run-time LMF configuration when the `txlink_rst_n` or `rxlink_rst_n` signals are asserted. Upon `txlink_rst_n` or `rxlink_rst_n` deassertion, the JESD204B IP core begins operation. The following sections describe the detailed operation for each subclass mode.

TX (Subclass 0)

Upon reset deassertion, the JESD204B TX IP core is in CGS phase. SYNC_N deassertion from the converter device enables the JESD204B TX IP core to exit CGS phase and enter ILAS phase (if `csr_lane_sync_en` = 1) or User Data phase (if `csr_lane_sync_en` = 0).

TX (Subclass 1)

Upon reset deassertion, the JESD204B TX IP core is in CGS phase. SYNC_N deassertion from the converter device enables the JESD204B TX IP core to exit CGS phase. The IP core ensures that at least one *SYSREF* rising edge is sampled before exiting CGS phase and entering ILAS phase. This is to prevent a race condition where the SYNC_N is deasserted before *SYSREF* is sampled. *SYSREF* sampling is crucial to ensure deterministic latency in the JESD204B Subclass 1 system.

TX (Subclass 2)

Similar to Subclass 1 mode, the JESD204B TX IP core is in CGS phase upon reset deassertion. The LMFC alignment between the converter and IP core starts after SYNC_N deassertion. The JESD204B TX IP core detects the deassertion of SYNC_N and compares the timing to its own LMFC. The required adjustment in the link clock domain is updated in the register map. You need to update the final phase adjustment

value in the registers for it to transfer the value to the converter during the ILAS phase. The DAC adjusts the LMFC phase and acknowledge the phase change with an error report. This error report contains the new DAC LMFC phase information, which allows the loop to iterate until the phase between them is aligned.

RX (Subclass 0)

The JESD204B RX IP core drives and holds SYNC_N (dev_sync_n signal) low when it is in reset. Upon reset deassertion, the JESD204B RX IP core checks if there is sufficient /K/ character to move its state machine out of synchronization request. Once sufficient /K/ character is detected, the IP core deasserts SYNC_N.

RX (Subclass 1)

The JESD204B RX IP core drives and holds the SYNC_N (dev_sync_n signal) low when it is in reset. Upon reset deassertion, the JESD204B RX IP core checks if there is sufficient /K/ character to move its state machine out of synchronization request. The IP core also ensures that at least one *SYSREF* rising edge is sampled before deasserting SYNC_N. This is to prevent a race condition where the SYNC_N is deasserted based on internal free-running LMFC count instead of the updated LMFC count after *SYSREF* is sampled.

RX (Subclass 2)

The JESD204B RX IP core behaves the same as in Subclass 1 mode. In this mode, the logic device is always the master timing reference. Upon SYNC_N deassertion, the ADC adjusts the LMFC timing to match the IP core.

4.3.6. Error Reporting Through SYNC_N Signal

The JESD204B TX IP core can detect error reporting through SYNC_N when SYNC_N is asserted for two frame clock periods (if F >= 2) or four frame clock periods (if F = 1). When the downstream device reports an error through SYNC_N, the TX IP core issues an interrupt. The TX IP core samples the SYNC_N pulse width using the link clock.

For a special case of F = 1, two frame clock periods are less than one link clock. Therefore, the error signaling from the receiver may be lost. You must program the converter device to extend the SYNC_N pulse to four frame clocks when F = 1.

The JESD204B RX IP core does not report an error through SYNC_N signaling. Instead, the RX IP core issues an interrupt when any error is detected.

You can check the csr_tx_err, csr_rx_err0, and csr_rx_err1 register status to determine the error types.

4.4. Clocking Scheme

This section describes the clocking scheme for the JESD204B IP core and transceiver.

Table 21. JESD204B IP Core Clocks

Clock Signal	Formula	Description
TX/RX Device Clock: `pll_ref_clk`	PLL selection during IP core generation	The PLL reference clock used by the TX Transceiver PLL or RX CDR. This is also the recommended reference clock to the PLL Intel FPGA IP core (for Arria V, Cyclone V, or Stratix V devices) or IOPLL Intel FPGA IP core (for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices).

TX/RX Link Clock:
`txlink_clk`
`rxlink_clk`
	Data rate/40	The timing reference for the JESD204B IP core.The link clock runs at data rate/40 because the IP core operates in a 32-bit data bus architecture after 8B/10B encoding.
		For Subclass 1, to avoid half link clock latency variation, you must supply the device clock at the same frequency as the link clock.
		The JESD204B transport layer in the design example requires both the link clock and frame clock to be synchronous.

TX/RX Frame Clock (in design example):
`txframe_clk`
`rxframe_clk`
	Data rate/(10 × F)	The frame clock as per the JESD204B specification. This clock is applicable to the JESD204B transport layer and other upstream devices that run in frame clock such as the PRBS generator/checker or any data processing blocks that run at the same rate as the frame clock.
		The JESD204B transport layer in the design example also supports running the frame clock in half rate or quarter rate by using the *FRAMECLK_DIV* parameter. The JESD204B transport layer requires both the link clock and frame clock to be synchronous. For more information, refer to the F1/F2_FRAMECLK_DIV parameter description and its relationship to the frame clock in the respective JESD204B Intel FPGA IP design example user guides.

TX/RX Transceiver Serial Clock and Parallel Clock	Internally derived from the data rate during IP core generation	The serial clock is the bit clock to stream out serialized data. The transceiver PLL supplies this clock and is internal to the transceiver. The parallel clock is for the transmitter PMA and PCS within the PHY. This clock is internal to the transceiver and is not exposed in the JESD204B IP core. For Arria V, Cyclone V, and Stratix V devices, these clocks are internally generated as the transceiver PLL is encapsulated within the JESD204B IP core's PHY. For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, you need to generate the transceiver PLL based on the data rate and connect the serial and parallel clock. You are recommended to select medium bandwidth for the transceiver PLL setting. These clocks are referred to as `*serial_clk` and `*bonding_clock` in Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices. Refer to the respective *Transceiver PHY IP Core User Guide* for more information.

TX/RX PHY Clock:
`txphy_clk`
`rxphy_clk`
	Data rate/40 (for all devices except Arria V GT/ST in PMA Direct mode)
	Data rate/80 (for Arria V GT/ST devices in PMA Direct mode)
		The PHY clock generated from the transceiver parallel clock for the TX path or the recovered clock generated from the CDR for the RX path.
		There is limited use for this clock. Avoid using this clock when PMA Direct mode is selected. Use this clock only if the JESD204B configuration is F=4 and the core is operating at Subclass 0 mode. This clock can be used as input for both the `txlink_clk` and `txframe_clk`, or `rxlink_clk` and `rxframe_clk`.
		When you set the PCS option to enable Hard PCS or Soft PCS mode, the `txphy_clk` connects to the transceiver `tx_std_clkout` signal and the `rxphy_clk` connects to the `rx_std_clkout` signal. These are the clock lines at the PCS and FPGA fabric interface. When you enable PMA Direct

continued...

Clock Signal	Formula	Description
		mode (for Arria V GT/ST only), the `txphy_clk` connects to the transceiver `tx_pma_clkout` signal and the `rxphy_clk` connects to the `rx_pma_clkout` signal. These are the clock lines at the PMA and PCS interface.

TX/RX AVS Clock:
`jesd204_tx_avs_clk`
`jesd204_rx_avs_clk`
	75–125 MHz	The configuration clock for the JESD204B IP core CSR through the Avalon-MM interface.

Transceiver Management Clock: `reconfig_clk`	100 MHz–125 MHz (Intel Arria 10) 100 MHz–125 MHz (Intel Cyclone 10 GX) 100 MHz–150 MHz (Intel Stratix 10)	The configuration clock for the transceiver CSR through the Avalon-MM interface. This clock is exported only when the transceiver dynamic reconfiguration option is enabled. This clock is only applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

Related Information

- Design Examples for JESD204B IP Core User Guide
  Provides information about design examples for Arria V, Cyclone V, Stratix V, and Intel Arria 10 devices using Intel Quartus Prime Standard Edition software.
- JESD204B Intel Arria 10 FPGA IP Design Example User Guide
- JESD204B Intel Stratix 10 FPGA IP Design Example User Guide
- JESD204B Intel Cyclone 10 GX FPGA IP Design Example User Guide

4.4.1. Device Clock

In a converter device, the sampling clock is typically the device clock.

For the JESD204B IP core in an FPGA logic device, you need one or two reference clocks as shown in Figure 21 on page 65 and Figure 22 on page 66. In the single reference clock design, the device clock is used as the transceiver PLL reference clock and also the core PLL reference clock. In the dual reference clock design, the device clock is used as the core PLL reference clock and the other reference clock is used as the transceiver PLL reference clock. The available frequency depends on the PLL type, bonding option, number of lanes, and device family. During IP core generation, the Intel Quartus Prime software recommends the available reference frequency for the transceiver PLL and core PLL based on user selection.

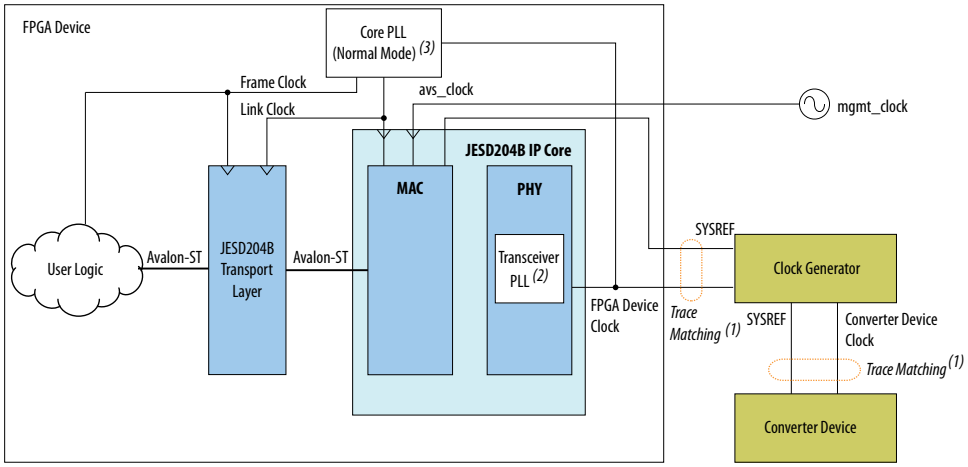
Note: Due to the clock network architecture in the FPGA, Intel recommends that you use the device clock to generate the link clock and use the link clock as the timing reference. You need to use the PLL Intel FPGA IP core (in Arria V, Cyclone V, and Stratix V devices) or IOPLL Intel FPGA IP core (in Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices) to generate the link clock and frame clock. The link clock is used in the JESD204B IP core (MAC) and the transport layer. You are recommended to supply the reference clock source through a dedicated reference clock pin.

Based on the JESD204B specification for Subclass 1, the device clock is the timing reference and is source synchronous with *SYSREF*. To achieve deterministic latency, match the board trace length of the SYSREF signal with the device clock. Maintain a constant phase relationship between the device clock and SYSREF signal pairs going to the FPGA and converter devices. Ideally, the SYSREF pulses from the clock generator should arrive at the FPGA and converter devices at the same time. To avoid half link clock latency variation, you must supply the device clock at the same frequency as the link clock.

The JESD204B protocol does not support rate matching. Therefore, you must ensure that the TX or RX device clock (*pll_ref_clk*) and the PLL reference clock that generates link clock (`txlink_clk` or `rxlink_clk`) and frame clock (`txframe_clk` or `rxframe_clk`) have 0 ppm variation. Both PLL reference clocks should come from the same clock chip.

Figure 21. JESD204B Subsystem with Shared Transceiver Reference Clock and Core Clock

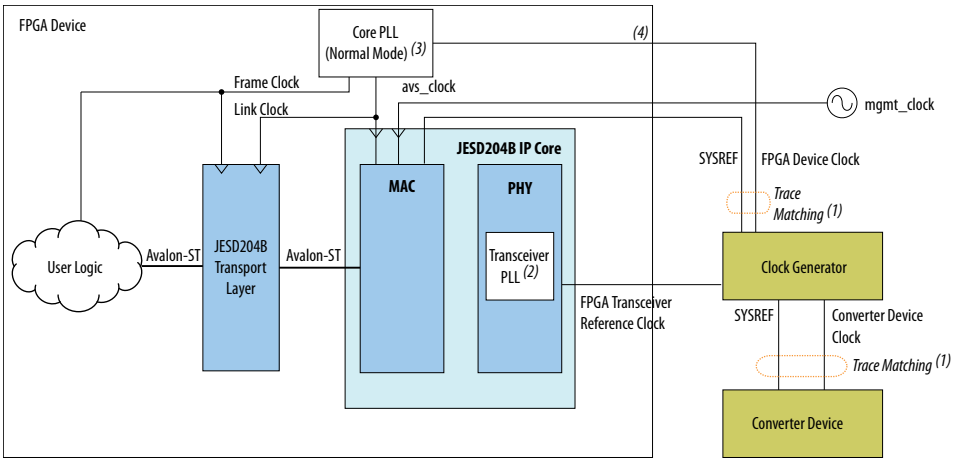
Note: Not applicable to Intel Stratix 10 devices.



Notes:

1. The device clock to the Intel core PLL and SYSREF must be trace matched. The device clock to the converter device and SYSREF must be trace matched. The phase offset between the SYSREF to the FPGA and converter devices should be minimal.
2. For Intel Arria 10 and Intel Cyclone 10 GX devices, the transceiver PLL is outside of the JESD204B IP core. For Arria V, Cyclone V, and Stratix V devices, the transceiver PLL is part of the JESD204B IP core.
3. The core PLL provides the link clock and frame clock. The link clock and frame clock must be synchronous. The AVS clock (e.g. mgmt_clk) can be asynchronous to the link and frame clock.

Figure 22. JESD204B Subsystem with Separate Transceiver Reference Clock and Core Clock



Notes:

1. The device clock to the Intel core PLL and SYSREF must be trace matched. The device clock to the converter device and SYSREF must be trace matched. The phase offset between the SYSREF to the FPGA and converter devices should be minimal.
2. For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, the transceiver PLL is outside of the JESD204B IP core. For Arria V, Cyclone V, and Stratix V devices, the transceiver PLL is part of the JESD204B IP core.
3. The core PLL provides the link clock and frame clock. The link clock and frame clock must be synchronous. The AVS clock (e.g. mgmt_clk) can be asynchronous to the link and frame clock.
4. You must use a dedicated reference clock input for the core PLL to compensate the FPGA clock network latency. This ensures that SYSREF is captured without any cycle variation.

Related Information

Clock Correlation on page 68

4.4.2. Link Clock

The device clock is the timing reference for the JESD204B system.

Due to the clock network architecture in the FPGA, JESD204B IP core does not use the device clock to clock the SYSREF signal because the GCLK or RCLK is not fully compensated. You are recommended to use the PLL Intel FPGA IP core (in Arria V, Cyclone V, and Stratix V devices) or IOPLL Intel FPGA IP core (in Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices) to generate both the link clock and frame clock. The PLL Intel FPGA IP core must operate in **normal mode** or **source synchronous mode** and uses a dedicated reference clock pin as the input reference clock source to achieve the following state:

- the GCLK and RCLK clock network latency is fully compensated.
- the link clock and frame clock at the registers are phase-aligned to the input of the clock pin.

To provide consistency across the design regardless of frame clock and sampling clock, the link clock is used as a timing reference.

The PLL Intel FPGA IP core should provide both the frame clock and link clock from the same PLL as these two clocks are treated as synchronous in the design.

For Subclass 0 mode, the device clock is not required to sample the SYSREF signal edge. The link clock does not need to be phase compensated to capture SYSREF. Therefore, you can generate both the link clock and frame clock using direct mode in

the PLL Intel FPGA IP core. If F = 4, where link clock is the same as the frame clock, you can use the parallel clock output from the transceiver (`txphy_clk` or `rxphy_clk` signal) except when the PCS option is in PMA Direct mode.

Related Information

Clock Correlation on page 68

4.4.3. Local MultiFrame Clock

The Local MultiFrame Clock (LMFC) is a counter generated from the link clock and depends on the F and K parameter.

The K parameter must be set between 1 to 32 and meet the requirement of at least a minimum of 17 octets and a maximum of 1024 octets in a single multiframe. In a 32-bit architecture, the K × F must also be in the order of four.

In a Subclass 1 deterministic latency system, the *SYSREF* frequency is distributed to the devices to align them in the system. The *SYSREF* resets the internal LMFC clock edge when the sampled *SYSREF* signal's rising edge transition from 0 to 1. Due to source synchronous signaling of *SYSREF* with respect to the device clock sampling (provided from the clock chip), the JESD204B IP core does not directly use the device clock to sample *SYSREF* but instead uses the link clock to sample *SYSREF*. Therefore, the Intel FPGA PLL IP core that provides the link clock must to be in **normal mode** to phase-compensate the link clock to the device clock.

Based on hardware testing, to get a fixed latency, at least 32 octets are recommended in an LMFC period so that there is a margin to tune the RBD release opportunity to compensate any lane-to-lane deskew across multiple resets. If F = 1, then K = 32 is optimal as it provides enough margin for system latency variation. If F = 2, then K = 16 and above (18/20/22/24/26/28/30/32) is sufficient to compensate lane-to-lane deskew.

The JESD204B IP core implements the local multiframe clock as a counter that increments in link clock counts. The local multiframe clock counter is equal to (F × K/4) in link clock as units. The rising edge of *SYSREF* resets the local multiframe clock counter to 0. There are two CSR bits that controls *SYSREF* sampling.

- `csr_sysref_singledet`—resets the local multiframe clock counter once and automatically cleared after *SYSREF* is sampled. This register also prevents CGS exit to bypass *SYSREF* sampling.
- `csr_sysref_alwayson`—resets the local multiframe clock counter at every rising edge of *SYSREF* that it detects. This register also enables the *SYSREF* period checker. If the provided *SYSREF* period violates the F and K parameter, an interrupt is triggered. However, this register does not prevent *CGS-SYSREF* race condition.

The following conditions occur if both CSR bits are set:

- resets the local multiframe clock counter at every rising edge of *SYSREF*.
- prevents *CGS-SYSREF* race condition.
- checks *SYSREF* period.

Related Information

Clock Correlation on page 68

4.4.4. Clock Correlation

This section describes the clock correlation between the device clock, link clock, frame clock, and local multiframe clock.

Example 1

Targeted device with LMF=222, K=16 and Data rate = 6.5 Gbps

Device Clock selected = 325 MHz (obtained during IP core generation)

Link Clock = 6.5 GHz/40 = 162.5 MHz Frame Clock = 6.5 GHz/(10x2) = 325 MHz

Local Multiframe clock = 325 MHz / 16 = 20.3125 MHz

SYSREF Frequency = Local Multiframe Clock / n; (n = integer; 1, 2, …)

Local multiframe clock counter = (F × K/4) = (2×16/4) = 8 link clocks [24]

Example 2

Targeted device with LMF=244, K=16 and Data rate = 5.0 Gbps

Device Clock selected = 125 MHz (obtained during IP core generation)

Link Clock = 5 GHz/40 = 125 MHz [25]

Frame Clock = 5 GHz /(10×4) = 125 MHz [25]

Local Multiframe clock = 125 MHz / 16 = 7.8125 MHz

SYSREF Frequency = Local Multiframe Clock / n; (n = integer; 1, 2, …)

Local multiframe clock counter = (F × K/4) = (4×16/4) = 16 link clocks [24]

Example 3

Targeted device with LMF=421, K=32 and Data rate = 10.0 Gbps

Device Clock selected = 250 MHz (obtained during IP core generation)

Link Clock = 10 GHz/40 = 250 MHz

Frame Clock = 10 GHz/(10×1) = 1 GHz [26]

Local Multiframe clock = 1 GHz / 32 = 31.25 MHz

---

(24) Eight link clocks means that the local multiframe clock counts from value 0 to 7 and then loopback to 0.

(25) The link clock and frame clock are running at the same frequency. You only need to generate one clock from the Intel FPGA PLL or Intel FPGA IO PLL IP core.

(26) In this example, the frame clock may not be able to run up to 1 GHz in the FPGA fabric. The JESD204B transport layer in the design example supports running the data stream of half rate (1 GHz/2 = 500 MHz), at two times the data bus width or of quarter rate (1GHz/4 = 250 MHz), at four times the data bus width.

SYSREF Frequency = Local Multiframe Clock / n; (n = integer; 1, 2, …)

Local multiframe clock counter = (F × K/4) = (1×32/4) = 8 link clocks [24]

Related Information

- Device Clock on page 64
- Link Clock on page 66
- Local MultiFrame Clock on page 67

4.5. Reset Scheme

All resets in the JESD204B IP core are synchronous reset signals and should be asserted and deasserted synchronously.

Note: Ensure that the resets are synchronized to the respective clocks for reset assertion and deassertion.

Table 22. JESD204B IP Core Resets

Reset Signal	Associated Clock	Description
`txlink_rst_n` `rxlink_rst_n`	TX/RX Link Clock	Active low reset. Intel recommends that you: • Assert the `txlink_rst_n`/`rxlink_rst_n` and `txframe_rst_n` /`rxframe_rst_n` signals when the transceiver is in reset. • Deassert the `txlink_rst_n` and `txframe_rst_n` signals after the Intel FPGA PLL IP core is locked and the `tx_ready[]` signal from the Transceiver Reset Controller is asserted. • Deassert the `rxlink_rst_n` and `rxframe_rst_n` signals after the Transceiver CDR `rx_islockedtodata[]` signal and `rx_ready[]` signal from the Transceiver Reset Controller are asserted. The `txlink_rst_n`/`rxlink_rst_n` and `txframe_rst_n` /`rxframe_rst_n` signals can be deasserted at the same time. These resets can only be deasserted after you configure the CSR registers.
`txframe_rst_n` `rxframe_rst_n`	TX/RX Frame Clock	Active low reset controlled by the clock and reset unit. If the TX/RX link clock and the TX/RX frame clock has the same frequency, both can share the same reset.
`tx_analogreset[L-1:0]` `rx_analogreset[L-1:0]`	Transceiver Native PHY Analog Reset	Active high reset controlled by the transceiver reset controller. This signal resets the TX/RX PMA. The link clock, frame clock, and AVS clock reset signals (`txlink_rst_n`/`rxlink_rst_n`, `txframe_rst_n`/`rxframe_rst_n` and `jesd204_tx_avs_rst_n`/`jesd204_rx_avs_rst_n`) can only be deasserted after the transceiver comes out of reset. [27]
`tx_analogreset_stat[L-1:0]` `rx_analogreset_stat[L-1:0]`	Transceiver Native PHY Analog Reset	TX PMA analog reset status port connected to the transceiver reset controller.[28]

continued...

(27) Refer to the *respective Transceiver PHY IP Core User Guide* for the timing diagram of the `tx_analogreset`, `rx_analogreset`, `tx_digitalreset`, and `rx_digitalreset` signals.

Reset Signal	Associated Clock	Description
		This signal is applicable for Intel Stratix 10 devices only.
tx_digitalreset[L-1:0] rx_digitalreset[L-1:0]	Transceiver Native PHY Digital Reset	Active high reset controlled by the transceiver reset controller. This signal resets the TX/RX PCS. The link clock, frame clock, and AVS clock reset signals (txlink_rst_n/rxlink_rst_n, txframe_rst_n/rxframe_rst_n and jesd204_tx_avs_rst_n/jesd204_rx_avs_rst_n) can only be deasserted after the transceiver comes out of reset. (27)
tx_digitalreset_stat[L-1:0] rx_digitalreset_stat[L-1:0]	Transceiver Native PHY Digital Reset	TX PCS digital reset status port connected to the transceiver reset controller.(28) This signal is applicable for Intel Stratix 10 devices only.
jesd204_tx_avs_rst_n jesd204_rx_avs_rst_n	TX/RX AVS (CSR) Clock	Active low reset controlled by the clock and reset unit. Typically, both signals can be deasserted after the core PLL and transceiver PLL are locked and out of reset. If you want to dynamically modify the LMF at run-time, you can program the CSRs after AVS reset is deasserted. This phase is referred to as the configuration phase. After the configuration phase is complete, then only the txlink_rst_n/rxlink_rst_n and txframe_rst_n/rxframe_rst_n signals can be deasserted.

Related Information

- V-Series Transceiver PHY User Guide
- Intel Arria 10 Transceiver PHY User Guide
- Intel Cyclone 10 GX Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide

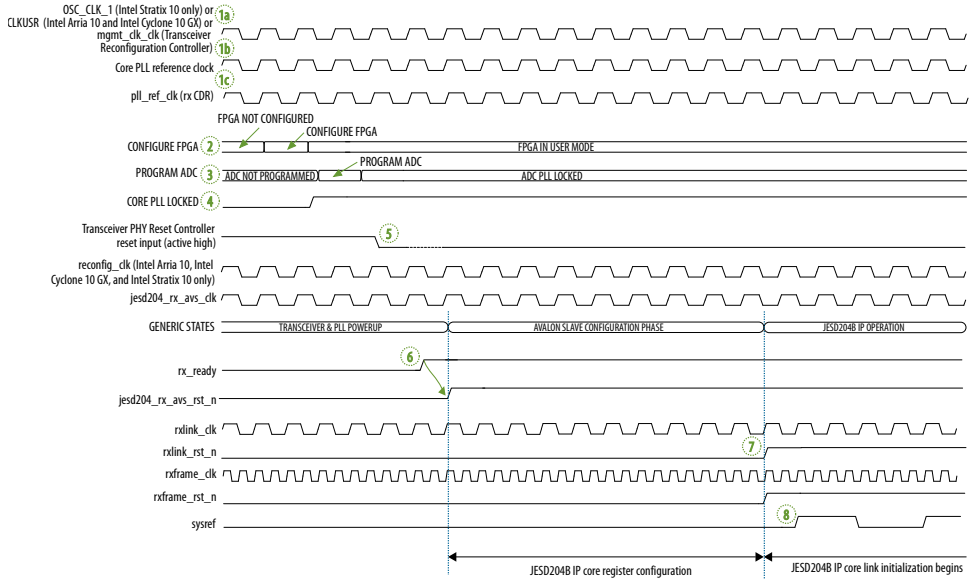
4.5.1. Reset Sequence

Intel recommends that you assert reset for the JESD204B IP core and transport layer when powering up the PLLs and transceiver.

(28) Refer to the respective *Intel Stratix 10 Transceiver PHY IP Core User Guide* for the timing diagram of the tx_analogreset_stat, rx_analogreset_stat, tx_digitalreset_stat, and rx_digitalreset_stat signals.

4.5.2. ADC–FPGA Subsystem Reset Sequence

Figure 23. ADC–FPGA Subsystem Reset Sequence Timing Diagram



The recommended ADC – FPGA subsystem bring-up sequence:

1. Provide a free-running and stable reference clock to the converter and FPGA in the JESD204B subsystem. The reference clock for the converter is the device clock. Intel recommends four reference clocks for the FPGA.

    a. The first reference clock is the calibration clock for the transceiver.

    - For Intel Stratix 10 devices, this is the clock at the OSC_CLK_1 pin for the calibration engine.
    - For Intel Arria 10 and Intel Cyclone 10 GX devices, this is the clock at the CLKUSR pin for the calibration engine.
    - For Arria V, Cyclone V, and Stratix V devices, this is the clock for the transceiver reconfiguration controller.

    b. The second reference clock is the management clock for the transceiver reconfiguration interface and the JESD204B IP core Avalon-MM interface.

    - If the dynamic reconfiguration option is enabled for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, this reference clock is connected to the `reconfig_clk` input port of the JESD204B IP core.

    c. The third reference clock is the transceiver reference clock.

    - For Intel Stratix 10, you must provide the reference clock at the transceiver dedicated reference clock input pin.
    - For Intel Arria 10, Intel Cyclone 10 GX, Arria V, Cyclone V, and Stratix V devices, this clock is also used as the reference clock for the core PLL (IOPLL Intel FPGA IP core for Intel Arria 10 and Intel Cyclone 10 GX devices; and PLL Intel FPGA IP core for Arria V, Cyclone V, and Stratix V devices) if you share the device clock and the transceiver reference clock (refer to Figure 21 on page 65).

d. The fourth reference clock is the core PLL reference clock (device clock).
   - For Intel Stratix 10, you must provide the reference clock at the dedicated reference clock input pin at the IO bank.
   - For Intel Arria 10, Intel Cyclone 10 GX, Arria V, Cyclone V, and Stratix V devices, this is the reference clock for the core PLL (IOPLL Intel FPGA IP core for Intel Arria 10 and Intel Cyclone 10 GX devices; and PLL Intel FPGA IP core for Arria V, Cyclone V, and Stratix V devices) if you do not share the device clock and the transceiver reference clock (refer to Figure 21 on page 65).

2. Configure the FPGA. Hold the RX transceiver channel in reset.
   - For Intel Arria 10 and Intel Cyclone 10 GX devices, if the reference clock is not available for the transceiver CDR before the FPGA is configured, you need to hold the RX transceiver channels in reset and perform user calibration for the RX transceiver channels after the reference clock is stable. For more information about user calibration for the transceiver channels, refer to the Calibration chapter in the *Intel Arria 10 or Intel Cyclone 10 GX Transceiver PHY User Guides*.
3. You can program the ADC through its SPI interface before or after configuring the FPGA. Ensure that the ADC PLL is locked before you proceed to the next step.
4. Ensure that the FPGA device clock core PLL is locked to the reference clock.
5. Deassert the FPGA RX transceiver channel reset. Do this by deasserting the reset input pin of the Transceiver PHY Reset Controller.
6. Once the transceiver is out of reset (the `rx_ready` signal from the Intel FPGA Transceiver PHY Reset Controller is asserted), deassert the Avalon-MM interface reset for the IP core. At the configuration phase, the subsystem can program the JESD204B IP core if the default IP core register settings need to change.
7. Deassert both the link reset for the IP core and the frame reset for the transport layer.
8. For subclass 1, if the continuous SYSREF pulses from the clock generator are present when the RX link reset is deasserted, the ADC-RX link initializes. If the SYSREF pulse is not present, trigger the clock generator to provide a SYSREF pulse to initialize the link. For subclass 0, the link initializes after the ADC is programmed and the RX link reset is deasserted.

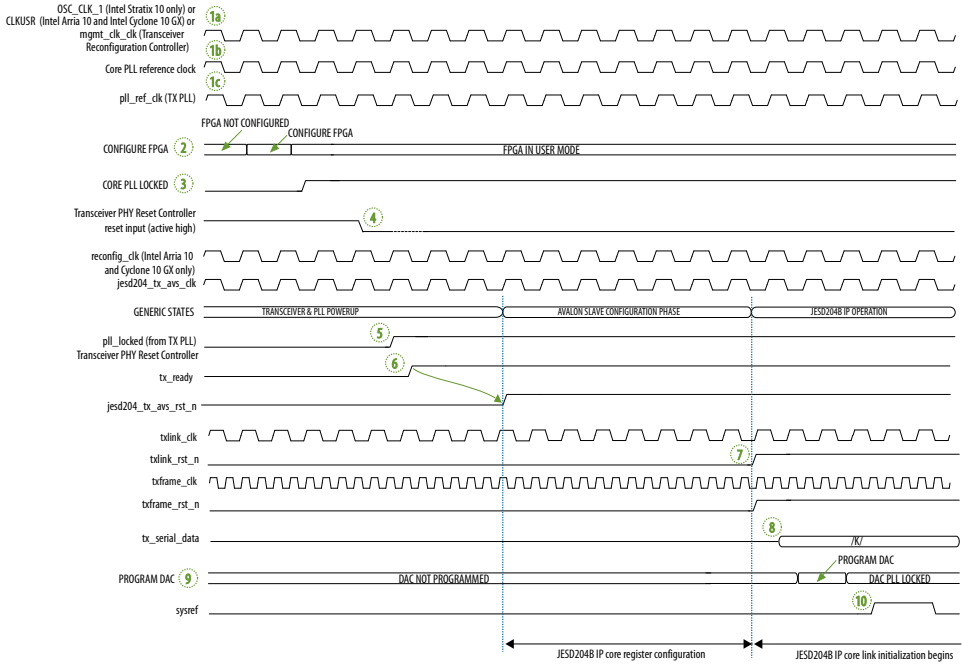
Related Information

- V-Series Transceiver PHY User Guide
- Intel Arria 10 Transceiver PHY User Guide
- Intel Cyclone 10 GX Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide
- V-Series Transceiver PHY User Guide
- Intel Arria 10 Transceiver PHY User Guide
- Intel Cyclone 10 GX Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide

- Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide

4.5.3. FPGA–DAC Subsystem Reset Sequence

Figure 24. FPGA–DAC Subsystem Reset Sequence Timing Diagram



The recommended FPGA – DAC subsystem bring-up sequence:

1. Provide a free-running and stable reference clock to the converter and FPGA in the JESD204B subsystem. The reference clock for the converter is the device clock. Intel recommends four reference clocks for the FPGA.

   a. The first reference clock is the calibration clock for the transceiver.

      - For Intel Stratix 10 devices, this is the clock at the OSC_CLK_1 pin for the calibration engine.
      - For Intel Arria 10 and Intel Cyclone 10 GX devices, this is the clock at the CLKUSR pin for the calibration engine.
      - For Stratix V, Arria V, and Cyclone V devices, this is the clock for the transceiver reconfiguration controller.

   b. The second reference clock is the management clock for the transceiver reconfiguration interface and the JESD204B IP core Avalon MM interface.

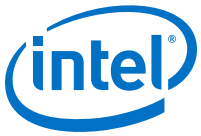
      - If the dynamic reconfiguration option is enabled for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, this reference clock is connected to the `reconfig_clk` input port of the JESD204B IP core.

   c. The third reference clock is the transceiver reference clock.

- For Intel Stratix 10, you must provide the reference clock at the transceiver dedicated reference clock input pin.
- For Intel Arria 10, Intel Cyclone 10 GX, Arria V, Cyclone V, and Stratix V, this clock is also used as the reference clock for the core PLL (IOPLL Intel FPGA IP core for Intel Arria 10 and Intel Cyclone 10 GX; and PLL Intel FPGA IP core for Arria V, Cyclone V, and Stratix V devices) if you share the device clock and the transceiver reference clock (refer to Figure 21 on page 65).

d. The fourth reference clock is the core PLL reference clock (device clock).
   - For Intel Stratix 10, you must provide the reference clock at the dedicated reference clock input pin at the IO bank.
   - For Intel Arria 10, Intel Cyclone 10 GX, Arria V, Cyclone V, and Stratix V, this is the reference clock for the core PLL (IOPLL Intel FPGA IP core for Intel Arria 10 and Intel Cyclone 10 GX devices; and PLL Intel FPGA IP core for Arria V, Cyclone V, and Stratix V devices) if you do not share the device clock and the transceiver reference clock (refer to Figure 22 on page 66).

2. Configure the FPGA. Hold the TX transceiver PLL and channel in reset.
   - For Intel Arria 10 and Intel Cyclone 10 GX devices, if the reference clock is not available for the transceiver PLL before the FPGA is configured, you need to hold the transceiver PLL and channels in reset and perform user calibration for the transceiver PLL and TX channels after the reference clock is stable. For more information about user calibration for the transceiver PLL and channels, refer to the *Calibration* chapter in the *Intel Arria 10 or Intel Cyclone 10 GX Transceiver PHY User Guides*.
3. Ensure that the FPGA device clock core PLL is locked to the reference clock.
4. Deassert the FPGA TX transceiver PLL and channel reset. Do this by deasserting the reset input pin of the Transceiver PHY Reset Controller.
5. Ensure that the FPGA transceiver PLL is locked to the reference clock.
6. Once the TX transceiver PLL and channel are out of reset (the `tx_ready` signal from the Transceiver PHY Reset Controller is asserted), deassert the Avalon-MM interface reset for the IP core. At the configuration phase, the subsystem can program the JESD204B IP core if the default IP core register settings need to change.
7. Deassert both the link reset for the IP core and the frame reset for the transport layer.
8. The TX IP core streams /K/ characters to the DAC after TX link reset is deasserted.
9. Program the DAC through its SPI interface.
10. For subclass 1, if the continuous SYSREF pulses from the clock generator are present when the TX link reset is deasserted, the TX-DAC link initializes. If the SYSREF pulse is not present, trigger the clock generator to provide a SYSREF pulse to initialize the link.
11. For subclass 0, the link initializes after the DAC is programmed and the TX link reset is deasserted.



4.6. Signals

The JESD204B IP core signals are listed by interface:

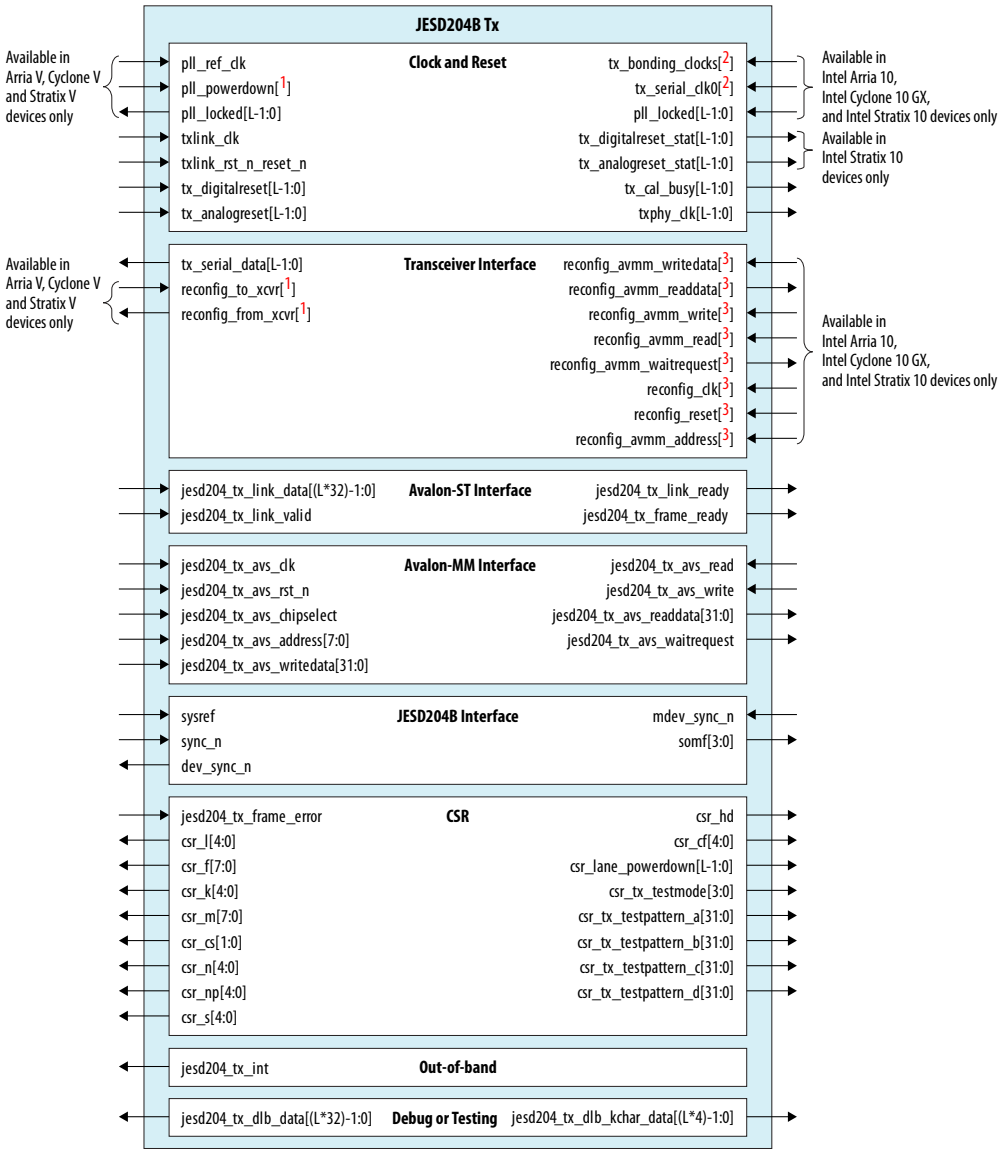
- Transmitter
- Receiver

Note: You should terminate any unused signals.

4.6.1. Transmitter

Figure 25. Transmitter Signal Diagram

L denotes the number of lanes.



Note:
1. Refer to the *Transmitter Signals* table for actual signal width.
2. Refer to the *Transmitter Signals* table for actual signal name.
3. Refer to the *Transmitter Signals* table for actual signal width and name.

Table 23. Transmitter Signals

Signal	Width	Direction	Description
Clocks and Resets			
pll_ref_clk	1	Input	Transceiver reference clock signal. The reference clock selection depends on the FPGA device family and data rate. This signal is only applicable for Arria V, Cyclone V, and Stratix V devices.
txlink_clk	1	Input	TX link clock signal. This clock is equal to the TX data rate divided by 40. For Subclass 1, you cannot use the output of txphy_clk signal as txlink_clk signal . To sample SYSREF correctly, the core PLL must provide the txlink_clk signal and must be configured as normal operating mode.
txlink_rst_n_reset_n	1	Input	Reset for the TX link clock signal. This reset is an active low signal.
txphy_clk[]	L	Output	TX parallel clock output for the TX transceiver with PCS option in Hard PCS or Soft PCS mode. This clock has the same frequency as txlink_clk signal. For PCS option in PMA Direct mode, this clock is half the frequency of txlink_clk signal. This clock is output as an optional port for user if the txlink_clk and txframe_clk signals are operating at the same frequency in Subclass 0 operating mode.
tx_digitalreset[] (29)	L	Input	Reset for the transceiver PCS block. This reset is an active high signal.
tx_digitalreset_stat[]	L	Output	TX PCS digital reset status port connected to the transceiver reset controller. This signal is applicable to Intel Stratix 10 devices only.
tx_analogreset[] (29)	L	Input	Reset for the transceiver PMA block. This reset is an active high signal.
tx_analogreset_stat[]	L	Output	TX PMA analog reset status port connected to the transceiver reset controller. This signal is applicable to Intel Stratix 10 devices only.
pll_locked[] (29)	L	Output	This is the PLL locked output signal for the hard transceiver of the Arria V, Cyclone V, and Stratix V devices. This signal is asserted to indicate that the TX transceiver PLL is locked.

continued...

(29) The Transceiver PHY Reset Controller IP core controls this signal.

Signal	Width	Direction	Description
		Input	This is the input signal for the Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10devices.
tx_cal_busy[] (29)	L	Output	TX calibration in progress signal. This signal is asserted to indicate that the TX transceiver calibration is in progress.
pll_powerdown[] (29)	• 1 if bonding mode = "xN" • L if bonding mode = feedback_compensation	Input	TX transceiver PLL power down signal. This signal is only applicable for Arria V, Cyclone V, and Stratix V devices.
tx_bonding_clocks (Single Channel) tx_bonding_clocks_ch<0..L-1>[] (Multiple Channels)	6	Input	The transceiver PLL bonding clocks. The transceiver PLL generation provides these clocks. This signal is only available if you select *Bonded* mode for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
tx_serial_clk0 (Single Channel) tx_serial_clk0_ch<0..L-1> (Multiple Channels)	1	Input	The transceiver PLL serial clock. This is the serializer clock in the PMA. The transceiver PLL generation provides these clocks. This signal is only available if you select *Non-bonded* mode for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

Signal	Width	Direction	Description
Transceiver Interface			
tx_serial_data[]	L	Output	Differential high speed serial output data. The clock is embedded in the serial data stream.
reconfig_to_xcvr[]	• (L+1)*70 if bonding mode = "xN" • L*140 if bonding mode = feedback compensation	Input	Reconfiguration signals from the Transceiver Reconfiguration Controller IP core to the PHY device. This signal is only applicable for Arria V, Cyclone V, and Stratix V devices. You must connect these signals to the Transceiver Reconfiguration Controller IP core regardless of whether run-time reconfiguration is enabled or disabled. The Transceiver Reconfiguration Controller IP core also supports various

continued...

Signal	Width	Direction	Description
			calibration function during transceiver power up.
reconfig_from_xcvr[]	• (L+1)*46 if bonding mode = "xN" • L*92 if bonding mode = feedback compensation	Output	Reconfiguration signals to the Transceiver Reconfiguration Controller IP core. This signal is only applicable for Arria V, Cyclone V, and Stratix V devices. You must connect these signals to the Transceiver Reconfiguration Controller IP core regardless of whether run-time reconfiguration is enabled or disabled. The Transceiver Reconfiguration Controller IP core also supports various calibration function during transceiver power up.
reconfig_clk reconfig_clk[] reconfig_clk_ch<0..L-1>	• 1 if **Share Reconfiguration Interface** = On • L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 1 bit per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Input	The Avalon-MM clock input. The frequency range is 100–125 MHz. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
reconfig_reset reconfig_reset[] reconfig_reset_ch<0..L-1>	• 1 if **Share Reconfiguration Interface** = On • L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 1 bit per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Input	Reset signal for the Transceiver Reconfiguration Controller IP core. This signal is active high and level sensitive. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

continued...

Signal	Width	Direction	Description
reconfig_avmm_address[] reconfig_avmm_address_ch<0..L-1>[]	Intel Arria 10 • log_2L*1024 if **Share Reconfiguration Interface** = On • 10*L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 10 bits per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On Intel Stratix 10 • log_2L*2048 if **Share Reconfiguration Interface** = On • 11*L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 11 bits per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Input	The Avalon-MM address. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
reconfig_avmm_writedata[] reconfig_avmm_writedata_ch<0..L-1>[]	• 32 if **Share Reconfiguration Interface** = On • 32*L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 32 bits per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Input	The input data. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
reconfig_avmm_readdata[] reconfig_avmm_readdata_ch<0..L-1>[]	• 32 if **Share Reconfiguration Interface** = On • 32*L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 32 bits per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Output	The output data. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

continued...

Signal	Width	Direction	Description

`reconfig_avmm_write`
`reconfig_avmm_write[]`
`reconfig_avmm_write_ch<0..L-1>`
	• 1 if **Share Reconfiguration Interface** = On
	• L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off
	• 1 bit per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On
		Input	Write signal. This signal is active high.
			This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

`reconfig_avmm_read`
`reconfig_avmm_read[]`
`reconfig_avmm_read_ch<0..L-1>`
	• 1 if **Share Reconfiguration Interface** = On
	• L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off
	• 1 bit per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On
		Input	Read signal. This signal is active high.
			This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

`reconfig_avmm_waitrequest`
`reconfig_avmm_waitrequest[]`
`reconfig_avmm_waitrequest_ch<0..L-1>`
	• 1 if **Share Reconfiguration Interface** = On
	• L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off
	• 1 bit per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On
		Output	Wait request signal.
			This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

Signal	Width	Direction	Description
Avalon-ST Interface			
`jesd204_tx_link_data[]`	`L*32`	Input	Indicates a 32-bit user data at `txlink_clk` clock rate, where four octets are packed into a 32-bit data width per lane. The data format is big endian. The first octet is located at bit[31:24], followed by bit[23:16], bit[15:8], and the last octet is bit[7:0]. Lane 0 data is always located in the lower 32-bit data. If more than one lane is instantiated, lane 1 is located at bit[63:32], with the first octet position at bit[63:56].

continued...

Signal	Width	Direction	Description
jesd204_tx_link_valid	1	Input	Indicates whether the data from the transport layer is valid or invalid. The Avalon-ST sink interface in the TX core cannot be backpressured and assumes that data is always valid on every cycle when the jesd204_tx_link_ready signal is asserted. • 0—data is invalid • 1—data is valid
jesd204_tx_link_ready	1	Output	Indicates that the Avalon-ST sink interface in the TX core is ready to accept data. The Avalon-ST sink interface asserts this signal on the JESD204B link state of *USER_DATA* phase. The ready latency is 0.
jesd204_tx_frame_ready	1	Output	Indicates that the Avalon-ST sink interface in the transport layer is ready to accept data. The Avalon-ST sink interface asserts this signal on the JESD204B link state of ILAS 4th multiframe and also the *USER_DATA* phase. The ready latency is 0.

Signal	Width	Direction	Description
Avalon-MM Interface			
jesd204_tx_avs_clk	1	Input	The Avalon-MM interface clock signal. This clock is asynchronous to all the functional clocks in the JESD204B IP core. The JESD204B IP core can handle any cross clock ratio and therefore the clock frequency can range from 75 MHz to 125 MHz.
jesd204_tx_avs_rst_n	1	Input	This reset is associated with the jesd204_tx_avs_clk signal. This reset is an active low signal. You can assert this reset signal asynchronously but must deassert it synchronously to the jesd204_tx_avs_clk signal. After you deassert this signal, the CPU can configure the CSRs.
jesd204_tx_avs_chipselect	1	Input	When this signal is present, the slave port ignores all Avalon-MM signals unless this signal is asserted. This signal must be used in combination with read or write. If the Avalon-MM bus does not support chip select, you are recommended to tie this port to 1.
jesd204_tx_avs_address[]	8	Input	For Avalon-MM slave, the interconnect translates the byte address into a word address in the address space so that each slave access is for a word of data. For example, address = 0 selects the first word of the slave and address = 1 selects the second word of the slave.
jesd204_tx_avs_writedata[]	32	Input	32-bit data for write transfers. The width of this signal and the jesd204_tx_avs_readdata[31:0] signal must be the same if both signals are present
jesd204_tx_avs_read	1	Input	This signal is asserted to indicate a read transfer. This is an active high signal and requires the jesd204_tx_avs_readdata[31:0] signal to be in use.

continued...

Signal	Width	Direction	Description
jesd204_tx_avs_write	1	Input	This signal is asserted to indicate a write transfer. This is an active high signal and requires the jesd204_tx_avs_writedata[31:0] signal to be in use.
jesd204_tx_avs_readdata[]	32	Output	32-bit data driven from the Avalon-MM slave to master in response to a read transfer.
jesd204_tx_avs_waitrequest	1	Output	This signal is asserted by the Avalon-MM slave to indicate that it is unable to respond to a read or write request. The JESD204B IP core ties this signal to 0 to return the data in the access cycle.

Signal	Width	Direction	Description
JESD204 Interface			
sysref	1	Input	*SYSREF* signal for JESD204B Subclass 1 implementation. For Subclass 0 and Subclass 2 mode, tie-off this signal to 0.
sync_n	1	Input	Indicates SYNC_N from the converter device or receiver. This is an active low signal and is asserted 0 to indicate a synchronization request or error reporting from the converter device. To indicate a synchronization request, the converter device must assert this signal for at least five frames and nine octets. To indicate an error reporting, the converter device must ensure that the pulse is at least one cycle of the txlink_clk signal or two cycles of the txframe_clk signal (whichever period is longer).
dev_sync_n	1	Output	Indicates a clean synchronization request. This is an active low signal and is asserted 0 to indicate a synchronization request only. The sync_n signal error reporting is being masked out of this signal. This signal is also asserted during software-initiated synchronization.
mdev_sync_n	1	Input	Indicates a multidevice synchronization request. Synchronize signal combination should be done externally and then input to the JESD204B IP core through this signal. • For subclass 0—combine the dev_sync_n signal from all multipoint links before connecting to the mdev_sync_n signal. • For subclass 1—connect the dev_sync_n signal to the mdev_sync_n signal for each link respectively. In a single link instance where multidevice synchronization is not needed, tie the dev_sync_n signal to this signal.
somf[]	4	Output	Indicates a start of multiframe. • [3]—start of multiframe for jesd204_tx_link_data[31:24] • [2]—start of multiframe for jesd204_tx_link_data[23:16] • [1]—start of multiframe for jesd204_tx_link_data[15:8] • [0]—start of multiframe for jesd204_tx_link_data[7:0]

Signal	Width	Direction	Description
CSR			
jesd204_tx_frame_error	1	Input	Optional signal to indicate an empty data stream due to invalid data. This signal is asserted high to indicate an error during data transfer from the transport layer to the TX core.
csr_l[]	5	Output	Indicates the number of active lanes for the link. The transport layer can use this signal as a run-time parameter.
csr_f[]	8	Output	Indicates the number of octets per frame. The transport layer can use this signal as a run-time parameter.
csr_k[]	5	Output	Indicates the number of frames per multiframe. The transport layer can use this signal as a run-time parameter.
csr_m[]	8	Output	Indicates the number of converters for the link. The transport layer can use this signal as a run-time parameter.
csr_cs[]	2	Output	Indicates the number of control bits per sample. The transport layer can use this signal as a run-time parameter.
csr_n[]	5	Output	Indicates the converter resolution. The transport layer can use this signal as a run-time parameter.
csr_np[]	5	Output	Indicates the total number of bits per sample. The transport layer can use this signal as a run-time parameter.
csr_s[]	5	Output	Indicates the number of samples per converter per frame cycle. The transport layer can use this signal as a run-time parameter.
csr_hd	1	Output	Indicates the high density data format. The transport layer can use this signal as a run-time parameter.
csr_cf[]	5	Output	Indicates the number of control words per frame clock period per link. The transport layer can use this signal as a run-time parameter.
csr_lane_powerdown[]	L	Output	Indicates which lane is powered down. You need to set this signal if you have configured the link and want to reduce the number of active lanes.
csr_tx_testmode[]	4	Output	Indicates the test mode for the JESD204B IP core and test pattern for the test pattern generator component in the design example. Refer to the tx_test register in the register map.
csr_tx_testpattern_a[]	32	Output	A 32-bit fixed data pattern for the test mode. (30)
csr_tx_testpattern_b[]	32	Output	A 32-bit fixed data pattern for the test mode. (30)
csr_tx_testpattern_c[]	32	Output	A 32-bit fixed data pattern for the test mode. (30)
csr_tx_testpattern_d[]	32	Output	A 32-bit fixed data pattern for the test mode. (30)

(30) You can use this signal in the transport layer to configure programmable test pattern.

Signal	Width	Direction	Description
Out-of-band (OOB)			
jesd204_tx_int	1	Output	Interrupt pin for the JESD204B IP core. Interrupt is asserted when any error or synchronization request is detected. Configure the tx_err_enable register to set the type of error that can trigger an interrupt.

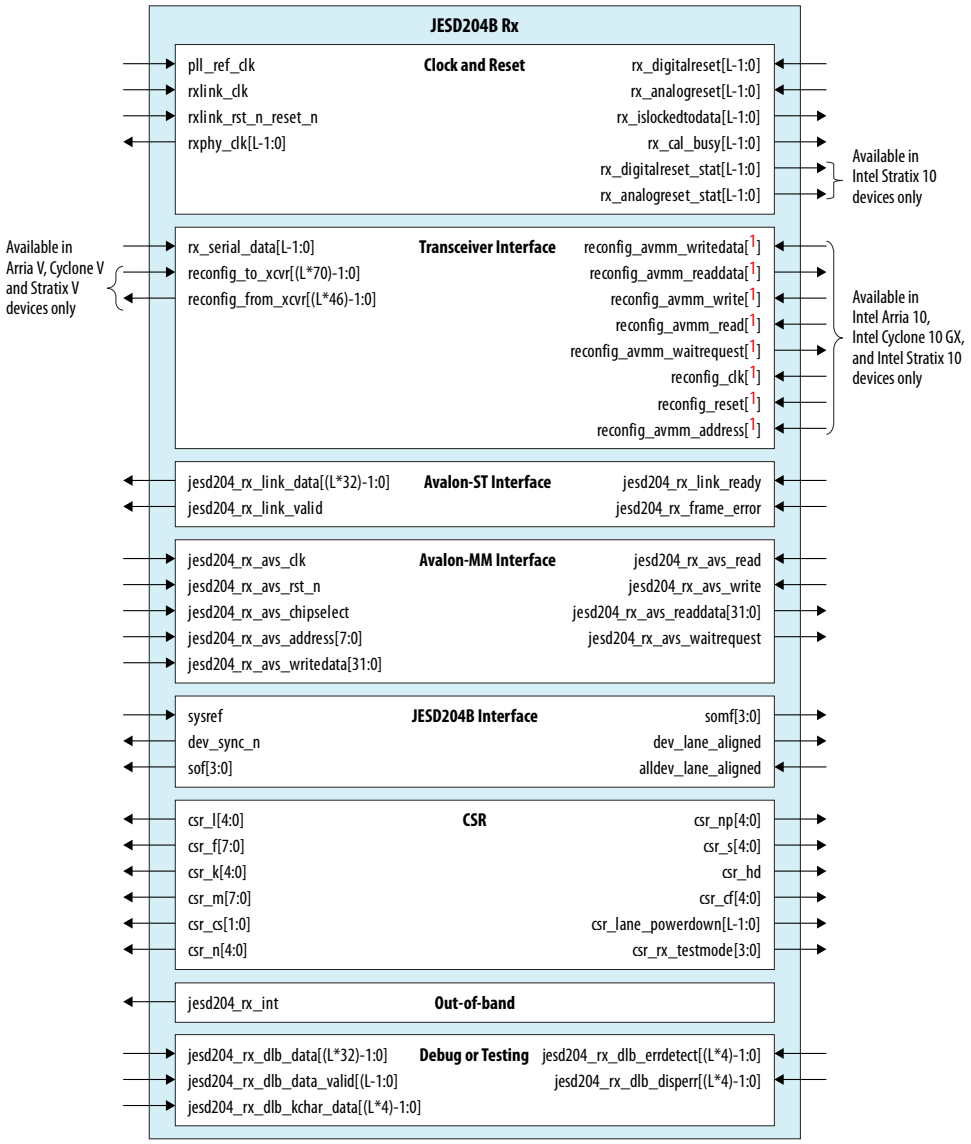
Signal	Width	Direction	Description
Debug or Testing			
jesd204_tx_dlb_data[]	L*32	Output	Optional signal for parallel data from the DLL in TX to RX loopback testing. (31)
jesd204_tx_dlb_kchar_data[]	L*4	Output	Optional signal to indicate the K character value for each byte in TX to RX loopback testing. (31)

(31) This signal is only for internal testing purposes. You can leave this signal disconnected.

4.6.2. Receiver

Figure 26. Receiver Signal Diagram

L denotes the number of lanes.



Note:
1. Refer to the *Receiver Signals* table for actual signal width and name.

Table 24. Receiver Signals

Signal	Width	Direction	Description
Clocks and Resets			
pll_ref_clk	1	Input	Transceiver reference clock signal.
rxlink_clk	1	Input	RX link clock signal used by the Avalon-ST interface. This clock is equal to RX data rate divided by 40.

continued...

Signal	Width	Direction	Description
			For Subclass 1, you cannot use the output of `rxphy_clk` signal as `rxlink_clk` signal. To sample SYSREF correctly, the core PLL must provide the `rxlink_clk` signal and must be configured as normal operating mode.
`rxlink_rst_n_reset_n`	1	Input	Reset for the RX link clock signal. This reset is an active low signal.
`rxphy_clk[]`	L	Output	Recovered clock signal. This clock is derived from the clock data recovery (CDR) and the frequency depends on the JESD204B IP core data rate. • For PCS option in Hard PCS or Soft PCS mode, this clock has the same frequency as the `rxlink_clk` signal. • For PCS option in PMA Direct mode, this clock is half the frequency of `rxlink_clk` signal.
`rx_digitalreset[]` (32)	L	Input	Reset for the transceiver PCS block. This reset is an active high signal.
`rx_digitalreset_stat[]`	L	Output	TX PCS digital reset status port connected to the transceiver reset controller.
`rx_analogreset[]` (32)	L	Input	Reset for the CDR and transceiver PMA block. This reset is an active high signal.
`rx_analogreset_stat[]`	L	Output	TX PMA analog reset status port connected to the transceiver reset controller.
`rx_islockedtodata[]` (32)	L	Output	This signal is asserted to indicate that the RX CDR PLL is locked to the RX data and the RX CDR has changed from LTR to LTD mode.
`rx_cal_busy[]` (32)	L	Output	RX calibration in progress signal. This signal is asserted to indicate that the RX transceiver calibration is in progress.

Signal	Width	Direction	Description
Transceiver Interface			
`rx_serial_data[]`	L	Input	Differential high speed serial input data. The clock is recovered from the serial data stream.
`reconfig_to_xcvr[]`	L*70	Input	Dynamic reconfiguration input for the hard transceiver. This signal is only applicable for for Arria V, Cyclone V, and Stratix V devices. You must connect these signals to the Transceiver Reconfiguration Controller IP core regardless of whether run-time

continued...

(32) The Transceiver PHY Reset Controller IP Core controls this signal.

Signal	Width	Direction	Description
			reconfiguration is enabled or disabled. The Transceiver Reconfiguration Controller IP core also supports various calibration function during transceiver power up.
reconfig_from_xcvr[]	L*46	Output	Dynamic reconfiguration output for the hard transceiver. This signal is only applicable for for Arria V, Cyclone V, and Stratix V devices You must connect these signals to the Transceiver Reconfiguration Controller IP core regardless of whether run-time reconfiguration is enabled or disabled. The Transceiver Reconfiguration Controller IP core also supports various calibration function during transceiver power up.
reconfig_clk reconfig_clk[] reconfig_clk_ch<0..L-1>	• 1 if **Share Reconfiguration Interface** = On • L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 1 bit per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Input	The Avalon-MM clock input. The frequency range is 100–125 MHz. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
reconfig_reset reconfig_reset[] reconfig_reset_ch<0..L-1>	• 1 if **Share Reconfiguration Interface** = On • L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 1 bit per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Input	Reset signal for the Transceiver Reconfiguration Controller IP core. This signal is active high and level sensitive. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

continued...

Signal	Width	Direction	Description
reconfig_avmm_address[] reconfig_avmm_address_ch<0..L-1>[]	Intel Arria 10 and Intel Cyclone 10 GX • log_2L*1024 if **Share Reconfiguration Interface** = On • 10*L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 10 bits per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On Intel Stratix 10 • log_2L*2048 if **Share Reconfiguration Interface** = On • 11*L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 11 bits per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Input	The Avalon-MM address. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
reconfig_avmm_writedata[] reconfig_avmm_writedata_ch<0..L-1>[]	• 32 if **Share Reconfiguration Interface** = On • 32*L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 32 bits per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Input	The input data. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
reconfig_avmm_readdata[] reconfig_avmm_readdata_ch<0..L-1>[]	• 32 if **Share Reconfiguration Interface** = On • 32*L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 32 bits per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Output	The output data. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

continued...

Signal	Width	Direction	Description
reconfig_avmm_write reconfig_avmm_write[] reconfig_avmm_write_ch<0..L-1>	• 1 if **Share Reconfiguration Interface** = On • L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 1 bit per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Input	Write signal. This signal is active high. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
reconfig_avmm_read reconfig_avmm_read[] reconfig_avmm_read_ch<0..L-1>	• 1 if **Share Reconfiguration Interface** = On • L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 1 bit per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Input	Read signal. This signal is active high. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
reconfig_avmm_waitrequest reconfig_avmm_waitrequest[] reconfig_avmm_waitrequest_ch<0..L-1>	• 1 if **Share Reconfiguration Interface** = On • L if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = Off • 1 bit per channel port if **Share Reconfiguration Interface** = Off and **Provide Separate Reconfiguration Interface for Each Channel** = On	Output	Wait request signal. This signal is only available if you enable dynamic reconfiguration for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

Signal	Width	Direction	Description
Avalon-ST Interface			
jesd204_rx_link_data[]	L*32	Output	Indicates a 32-bit data from the DLL to the transport layer. The data format is big endian, where the earliest octet is placed in bit [31:24] and the latest octet is placed in bit [7:0].
jesd204_rx_link_valid	1	Output	Indicates whether the data to the transport layer is valid or invalid. The Avalon-ST source interface in the RX core cannot be backpressured and transmits the data when the jesd204_rx_data_valid signal is asserted. • 0—data is invalid • 1—data is valid
jesd204_rx_link_ready	1	Input	Indicates that the Avalon-ST sink interface in the transport layer is ready to receive data.
jesd204_rx_frame_error	1	Input	Indicates an empty data stream due to invalid data. This signal is asserted high to indicate an error during data transfer from the RX core to the transport layer.

Signal	Width	Direction	Description
Avalon-MM Interface			
jesd204_rx_avs_clk	1	Input	The Avalon-MM interface clock signal. This clock is asynchronous to all the functional clocks in the JESD204B IP core. The JESD204B IP core can handle any cross clock ratio and therefore the clock frequency can range from 75 MHz to 125 MHz.
jesd204_rx_avs_rst_n	1	Input	This reset is associated with the jesd204_rx_avs_clk signal. This reset is an active low signal. You can assert this reset signal asynchronously but must deassert it synchronously to the jesd204_rx_avs_clk signal. After you deassert this signal, the CPU can configure the CSRs.
jesd204_rx_avs_chipselect	1	Input	When this signal is present, the slave port ignores all Avalon-MM signals unless this signal is asserted. This signal must be used in combination with read or write. If the Avalon-MM bus does not support chip select, you are recommended to tie this port to 1.
jesd204_rx_avs_address[]	8	Input	For Avalon-MM slave, the interconnect translates the byte address into a word address in the address space so that each slave access is for a word of data. For example, address = 0 selects the first word of the slave and address = 1 selects the second word of the slave.
jesd204_rx_avs_writedata[]	32	Input	32-bit data for write transfers. The width of this signal and the jesd204_rx_avs_readdata[31:0] signal must be the same if both signals are present.
jesd204_rx_avs_read	1	Input	This signal is asserted to indicate a read transfer. This is an active high signal and requires the jesd204_rx_avs_readdata[31:0] signal to be in use.
jesd204_rx_avs_write	1	Input	This signal is asserted to indicate a write transfer. This is an active high signal and requires the jesd204_rx_avs_writedata[31:0] signal to be in use.
jesd204_rx_avs_readdata[]	32	Output	32-bit data driven from the Avalon-MM slave to master in response to a read transfer.
jesd204_rx_avs_waitrequest	1	Output	This signal is asserted by the Avalon-MM slave to indicate that it is unable to respond to a read or write request. The JESD204B IP core ties this signal to 0 to return the data in the access cycle.

Signal	Width	Direction	Description
JESD204 Interface			
sysref	1	Input	SYSREF signal for JESD204B Subclass 1 implementation. For Subclass 0 and Subclass 2 mode, tie-off this signal to 0.
dev_sync_n	1	Output	Indicates a *SYNC~* from the receiver. This is an active low signal and is asserted 0 to indicate a synchronization request. Instead of reporting the link error through this signal, the JESD204B IP core uses the jesd204_rx_int signal to interrupt the CPU. For multilink synchronization, you can optionally connect the DEV_SYNC_N from each IP core to the input of an AND gate. The output of the AND gate is

continued...

Signal	Width	Direction	Description
			exported to the FPGA pins for connection to the analog-to-digital converters. Refer to AN803 and AN804 for more information about the connection guidelines.
sof[]	4	Output	Indicates a start of frame. • [3]—start of frame for jesd204_rx_link_data[31:24] • [2]—start of frame for jesd204_rx_link_data[23:16] • [1]—start of frame for jesd204_rx_link_data[15:8] • [0]—start of frame for jesd204_rx_link_data[7:0]
somf[]	4	Output	Indicates a start of multiframe. • [3]—start of multiframe for jesd204_rx_link_data[31:24] • [2]—start of multiframe for jesd204_rx_link_data[23:16] • [1]—start of multiframe for jesd204_rx_link_data[15:8] • [0]—start of multiframe for jesd204_rx_link_data[7:0]
dev_lane_aligned	1	Output	Indicates that all lanes for this device are aligned.
alldev_lane_aligned	1	Input	Aligns all lanes for this device. For multidevice synchronization, input all the dev_lane_aligned signals to an AND gate and connect the AND gate output to this pin. For single device support, connect the dev_lane_aligned signal back to this signal.

Signal	Width	Direction	Description
CSR			
csr_l[]	5	Output	Indicates the number of active lanes for the link. The transport layer can use this signal as a run-time parameter.
csr_f[]	8	Output	Indicates the number of octets per frame. The transport layer can use this signal as a run-time parameter.
csr_k[]	5	Output	Indicates the number of frames per multiframe. The transport layer can use this signal as a run-time parameter.
csr_m[]	8	Output	Indicates the number of converters for the link. The transport layer can use this signal as a run-time parameter.
csr_cs[]	2	Output	Indicates the number of control bits per sample. The transport layer can use this signal as a run-time parameter.
csr_n[]	5	Output	Indicates the converter resolution. The transport layer can use this signal as a run-time parameter.
csr_np[]	5	Output	Indicates the total number of bits per sample. The transport layer can use this signal as a run-time parameter.

continued...

Signal	Width	Direction	Description
csr_s[]	5	Output	Indicates the number of samples per converter per frame cycle. The transport layer can use this signal as a run-time parameter.
csr_hd	1	Output	Indicates the high density data format. The transport layer can use this signal as a run-time parameter.
csr_cf[]	5	Output	Indicates the number of control words per frame clock period per link. The transport layer can use this signal as a run-time parameter.
csr_lane_powerdown[]	L	Output	Indicates which lane is powered down. You need to set this signal if you have configured the link and want to reduce the number of active lanes.
csr_rx_testmode[]	4	Output	Indicates the test mode for the JESD204B IP core and test pattern for the test pattern checker component in the design example. Refer to the rx_test register in the register map.

Signal	Width	Direction	Description
Out-of-band (OOB)			
jesd204_rx_int	1	Output	Interrupt pin for the JESD204B IP core. Interrupt is asserted when any error is detected. Configure the rx_err_enable register to set the type of error that can trigger an interrupt.

Signal	Width	Direction	Description
Debug or Testing			
jesd204_rx_dlb_data[]	L*32	Input	Optional signal for parallel data to the DLL in TX to RX loopback testing. (33)
jesd204_rx_dlb_data_valid[]	L	Input	Optional signal to indicate valid data for each byte in TX to RX loopback testing. (33)
jesd204_rx_dlb_kchar_data[]	L*4	Input	Optional signal to indicate the K character value for each byte in TX to RX loopback testing. (33)
jesd204_rx_dlb_errdetect[]	L*4	Input	Optional signal to indicate 8B/10B error. (33)
jesd204_rx_dlb_ disperr[]	L*4	Input	Optional signal to indicate running disparity. (33)

Related Information

- AN803: Implementing ADC-Intel Arria 10 Multi-Link Design with JESD204B RX IP Core
- AN804: Implementing ADC-Intel Stratix 10 Multi-Link Design with JESD204B RX IP Core

(33) This signal is only for internal testing purposes. Tie this signal to low.

4.7. Registers

The JESD204B IP core supports a basic one clock cycle transaction bus. There is no support for burst mode and wait-state feature (the `avs_waitrequest` signal is tied to 0). The JESD204B IP core Avalon-MM slave interface has a data width of 32 bits and is implemented based on word addressing. The Avalon-MM slave interface does not support byte enable access.

Each write transfer has a *writeWaitTime* of 0 cycle while a read transfer has a *readWaitTime* of 1 cycle and *readLatency* of 1 cycle.

The following HTML files list the TX and RX core registers. The register address in the register map is written based on byte addressing. The Platform Designer interconnect automatically converts from byte to word addressing. You do not need to manually shift the address bus. If the Avalon-MM master interfaces to the IP core Avalon-MM slave without the Platform Designer interconnect, to perform byte to word addressing conversion, you are recommended to shift the Avalon-MM master address bus by 2 bits (divide by 4) when connecting to the IP core's Avalon-MM slave. In this connection, the Avalon-MM master address bit[2] connects to the IP core (Avalon-MM slave) address bit[0], while the Avalon-MM master bit[9] connects to the IP core address bit[7].

Note: For Intel Stratix 10 devices, run-time access for certain registers have been disabled. Refer to the TX and RX register map for more information.

Related Information

- JESD204B RX Address Map and Register Definitions
- JESD204B TX Address Map and Register Definitions

4.7.1. Register Access Type Convention

This table describes the register access type for Intel FPGA IP cores.

Table 25. Register Access Type and Definition

Access Type	Definition
RO	Software read only (no effect on write). The value is hard-tied internally to either '0' or '1' and does not vary.
RO/v	Software read only (no effect on write). The value may vary.
RC	• Software reads and returns the current bit value, then the bit is self-clear to 0. • Software reads also cause the bit value to be cleared to 0.

continued...

Access Type	Definition
RW	• Software reads and returns the current bit value. • Software writes and sets the bit to the desired value.
RW1C	• Software reads and returns the current bit value. • Software writes 0 and have no effect. • Software writes 1 and clear the bit to 0, if the bit has been set to 1 by hardware. • Hardware sets the bit to 1. • Software clear has higher priority than hardware set.
RW1S	• Software reads and returns the current bit value. • Software writes 0 and have no effect. • Software writes 1 and set the bit to 1. • Hardware clears the bit to 0, if the bit has been set to 1 by software. • Software set has higher priority than hardware clear.

5. JESD204B IP Core Deterministic Latency Implementation Guidelines

Subclass 1 and Subclass 2 modes support deterministic latency. This section describes the features available in the JESD204B IP core that you can use to achieve Subclass 1 deterministic latency in your design. This section also covers some best practices for Subclass 1 implementation like constraining the incoming SYSREF signal and maintaining deterministic latency during link reinitialization.

Features available:

- Programmable RBD offset.
- Programmable LMFC offset.

5.1. Constraining Incoming SYSREF Signal

The SYSREF signal resets the LMFC counter in the IP core for subclass 1 implementation. Constraining the SYSREF signal ensures that the setup relationship between SYSREF and device clock is established.

The setup time is analyzed when you set the timing constraint for the SYSREF signal in the user `.sdc` file. When the setup time is met, the SYSREF signal detection by the IP core is deterministic; the number of link clock cycles of SYSREF signal that arrives at the FPGA pin to the LMFC counter resets, is deterministic.

Apply the `set_input_delay` constraint on the SYSREF signal with respect to device clock in the user `.sdc` file:

```
set_input_delay -clock <device clock name at FPGA pin> <sysref IO delay
in ns> [get_ports <sysref name at FPGA pin >]
```

The SYSREF IO delay is the board trace length mismatch between device clock and SYSREF. For example:

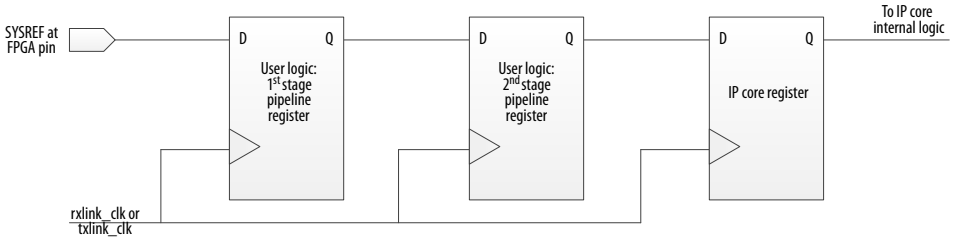
```
set_input_delay -clock device_clk 0.5 [get_ports sysref]
```

The above statement constrains the FPGA SYSREF signal (*sysref*), with respect to the FPGA device clock (*device_clk*) pin. The trace length mismatch resulted in 500 ps or 0.5 ns difference in time arrival at the FPGA pins between SYSREF and device clock.

In most cases, the register in the IP core, which detects the SYSREF signal, is far away from the SYSREF I/O pin. The long interconnect routing delay results in timing violation. You are recommended to use multi-stages pipeline registers to close timing. Use the same clock domain as the JESD204B IP core's `rxlink_clk` and `txlink_clk` to clock the multi-stages pipeline registers.

Figure 27. Multi-Stage Pipeline Register for SYSREF Signal

Figure shows a two stages pipeline registers for the SYSREF signal.



5.2. Programmable RBD Offset

In the RX IP core, the programmable RBD offset provides flexibility for an early RBD release to optimize the latency through the IP core. You can configure the RBD offset using the *csr_rbd_offset* field in the `syncn_sysref_ctrl` register.

You must set a safe RBD offset value to ensure deterministic latency from one power cycle to another power cycle. Follow these steps to set a safe RBD offset value:

1. Read the RBD count from the *csr_rbd_count* field in `rx_status0` register. Record the value.
2. Power cycle the JESD204B subsystem, which consists of the FPGA and converter devices.
3. Read the RBD count again and record the value.
4. Repeat steps 1 to 3 at least 5 times and record the RBD count values.
5. Set the *csr_rbd_offset* accordingly with one LMFC count tolerance.
6. Perform multiple power cycles and make sure lane deskew error does not occur using this RBD offset value.

The RBD count must be fairly consistent, within 2 counts variation from one power cycle to another power cycle. In the following examples, the parameter values are L > 1, F=1 and K=32. The legal values of the LMFC counter is 0 to ((FxK/4)-1), which is 0 to 7. In Figure 28 on page 98 , the latest arrival lane variation falls within 1 local multiframe period. In this scenario, if latency is not a concern, you can leave the default value of *csr_rbd_offset*=0, which means the RBD elastic buffer is released at the LMFC boundary. In Figure 29 on page 99 , the latest arrival lane variation spans across 2 local multiframes; the latest arrival lane variation happens before and after the LMFC boundary. In this scenario, you need to configure the RBD offset correctly to avoid lane deskew error as indicated in bit 4 of `rx_err0` register.

Figure 28. Early RBD Release Opportunity for Latest Arrival Lane Within One Local Multi Frame Scenario

In this example, the SYSREF pulse at *rx_sysref* port of the IP core is sampled by the internal register. After 2 link clock cycles, the LMFC counter resets. The delay from SYSREF sampled high to LMFC counter resets is deterministic. The transition of /K/ character to /R/ character marks the beginning of ILAS phase. The number of LMFC count of the /R/ character relative to the next LMFC boundary in the latest arrival lane is reported as the RBD count. In the first power cycle, the /R/ character is received at 4 LMFC counts before the next LMFC boundary, hence the RBD count = 4. In the second power cycle, the /R/ character is received at 3 LMFC counts before next LMFC boundary, hence the RBD count = 3. In five power cycles, the RBD count varies from 3 to 5. Since there are limited number of power cycles and boards for characterization, 1 LMFC count tolerance is allocated as a guide to set early RBD release opportunity. Hence, setting *csr_rbd_offset* = 1 can safely release the elastic buffer 1 LMFC count earlier at LMFC count 7 before the next LMFC boundary. A lane de-skew error occurs when the RBD elastic buffer is released before the latest arrival lane.

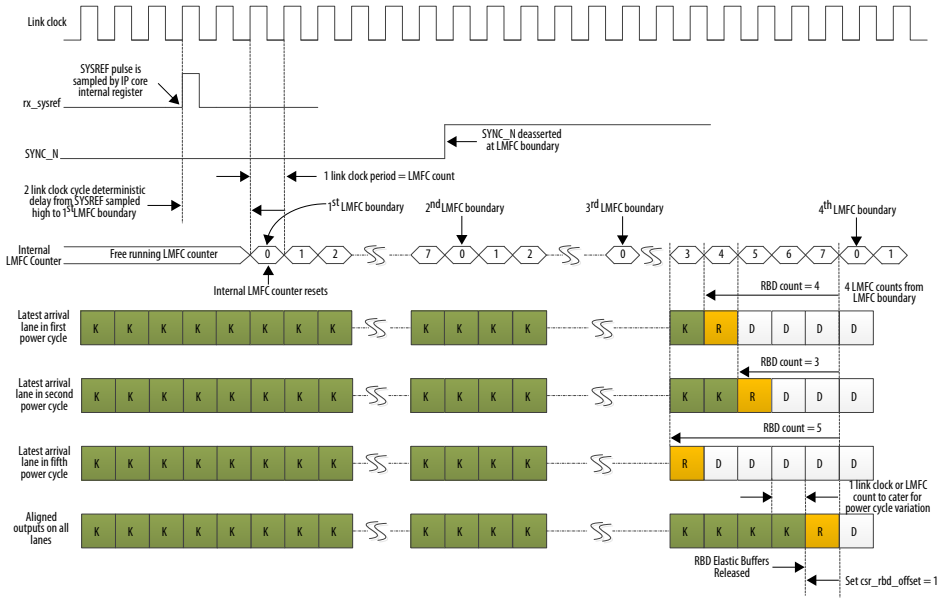
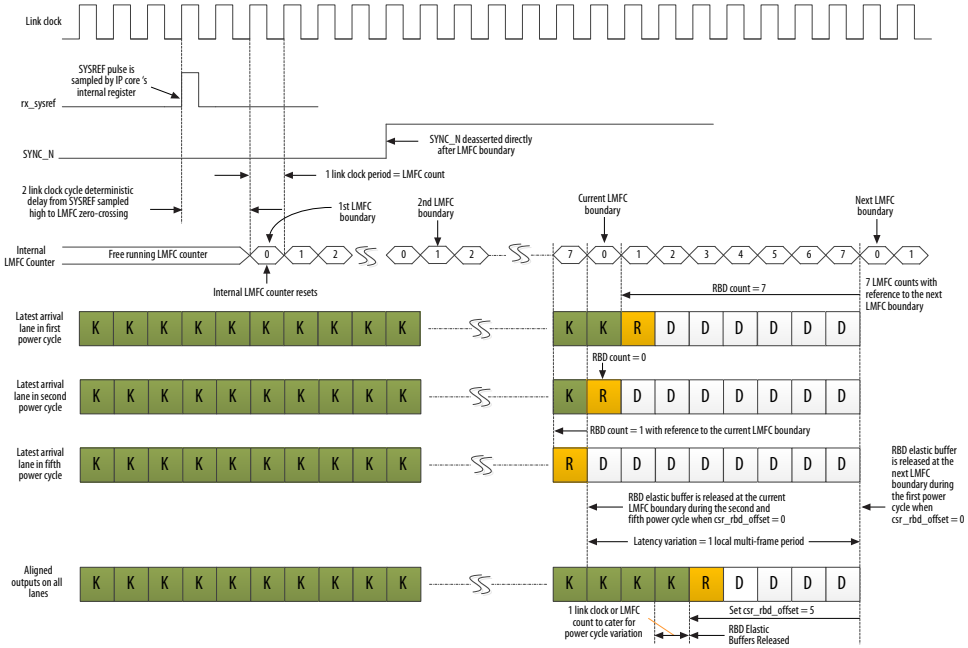


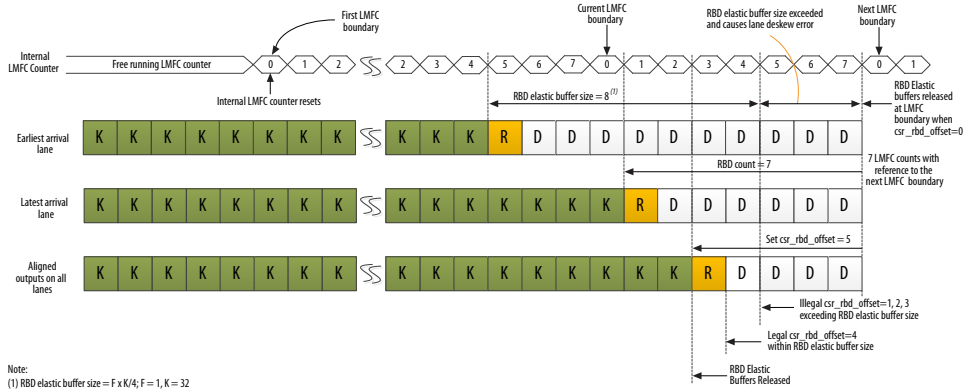
Figure 29. Early RBD Release Opportunity for Latest Arrival Lane Across Two Local Multi Frames Scenario

In this example, the RBD count varies from 7 to 1; the /R/ character is received at the previous local multiframe when the RBD count = 1; the /R/ character is received at the current local multiframe when the RBD count = 0 and 7. In this scenario, deterministic latency is not guaranteed because the RBD elastic buffer is released either at the current LMFC boundary when the RBD count = 0 and 1, or one local multiframe period later at the next LMFC boundary when the RBD count = 7. You can fix this issue by setting the RBD offset so that the RBD elastic buffer is always released at the next local multiframe. Setting *csr_rbd_offset* = 5 forces the release of RBD elastic buffer 5 LMFC counts before the next LMFC boundary. This corresponds to LMFC count of 3 at the current local multiframe. In this scenario, setting *csr_rbd_offset* not only optimizes user data latency through the IP core, it also resolves the deterministic latency issue.



In the example above, lane deskew error happens if the sum of the difference of /R/ character's LMFC count in the earliest arrival lane to the latest arrival lane, and the number of LMFC count up to the release of RBD elastic buffer exceeds the RBD elastic buffer size. If this is the root cause of lane deskew error, setting RBD offset is one of the techniques to overcome this issue. Not every RBD offset value is legal. Figure below illustrates the technique to decide the legal RBD offset value.

Figure 30. Selecting Legal RBD Offset Value



Because the IP core does not report the position of the earliest lane arrival with respect to the LMFC boundary, you must perform multiple power cycles to observe the RBD count and tune the RBD offset accordingly until no lane deskew error occurs. From the example in the figure above, the recommended RBD offset value is 4 or 5. Setting RBD offset to 1, 2 or 3 is illegal because this exceeds the RBD elastic buffer size for the F and K configurations.

Related Information

SYNC_N Signal on page 59

5.3. Programmable LMFC Offset

If your JESD204B subsystem design has deterministic latency issue, the programmable LMFC offset in the TX and RX IP cores provides flexibility to ensure that deterministic latency can be achieved.

The TX LMFC offset can align the TX LMFC counter to the LMFC counter in DAC; the RX LMFC offset can align the RX LMFC counter to the LMFC counter in ADC. Phase offset between the TX and RX LMFC counters in the both ends of the JESD204B link contributes to deterministic latency uncertainty. The phase offset is caused by:

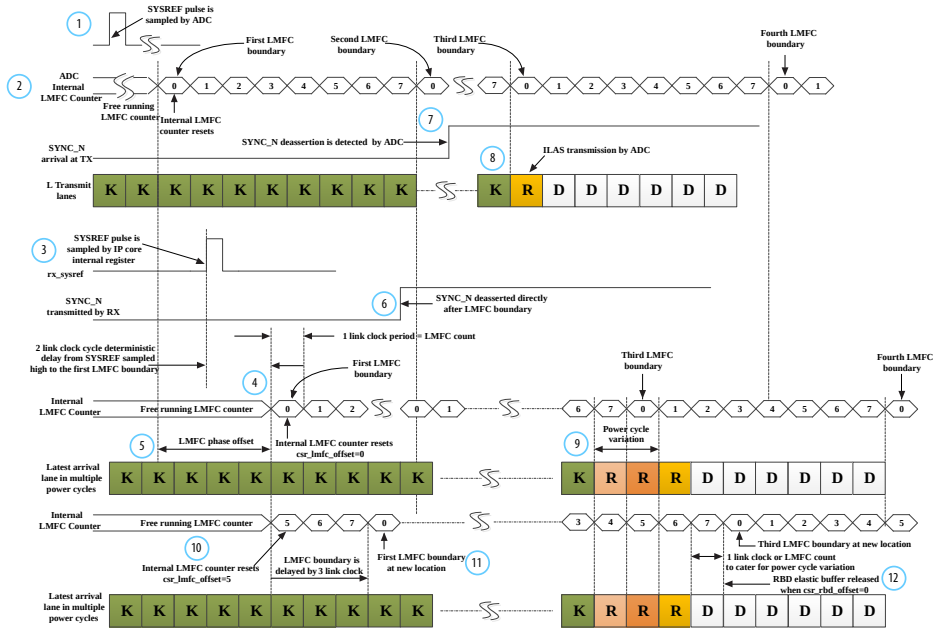
- SYSREF trace length mismatch in the PCB between the TX and RX devices (FPGA and converters).
- delay differences in resetting the LMFC counter when SYSREF pulses are detected by the FPGA and converter devices.

The RX device in the JESD204B link is responsible for deterministic latency adjustments. The following figure illustrates the adjustments that you can make to the RX LMFC offset using the *csr_lmfc_offset* field in the `syncn_sysref_ctrl` register. This is an alternative to using *csr_rbd_offset* to achieve deterministic latency.

Figure 31. Selecting Legal LMFC Offset Value for RX

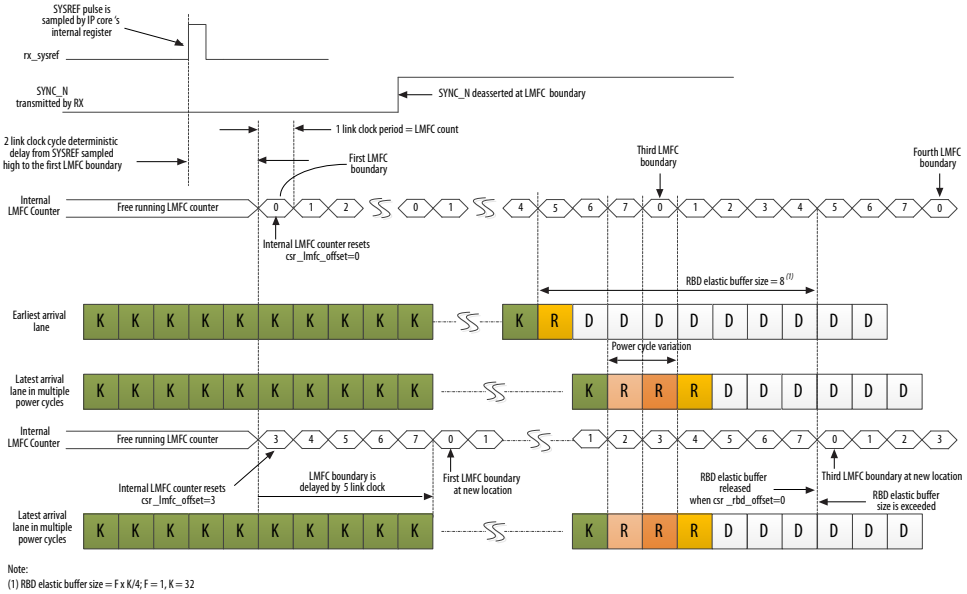
Sequence of events in the diagram:

1. Due to trace length mismatch, SYSREF pulse arrives at the ADC first.
2. Some deterministic delay occurs in between the time when the SYSREF pulse is sampled high to the reset of the ADC internal LMFC counter.
3. The SYSREF pulse arrives at the FPGA IP core port, `rx_sysref`, after the pulse's arrival at the ADC.
4. The FPGA IP core's internal LMFC counter resets two link clock cycles after SYSREF is sampled.
5. The LMFC phase offset between the LMFC counter at ADC and FPGA is ~3.5 link clock cycles.
6. The FPGA deasserts SYNC_N at the LMFC boundary.
7. The ADC JESD204B core detects the SYNC_N deassertion.
8. Because SYNC_N deassertion is detected after the second LMFC boundary at ADC, ILAS transmission begins at the third LMFC boundary.
9. In this example, the ILAS arrives at the IP core's RBD elastic buffer within one local multiframe. In other system, the arrival at the RBD elastic buffer can span more than one local multiframe. Assuming *csr_rbd_offset* = 0, RBD elastic buffer may be released at the third or fourth LMFC boundary due to power cycle variation.
10. Setting *csr_lmfc_offset* = 5 resets the LMFC counter to the value of 5.
11. The first LMFC boundary is delayed by three link clock cycles.
12. The third LMFC boundary has been delayed past the latest arrival lane power cycle variation. The RBD elastic buffer is always released at the third LMFC boundary.



You should set a safe LMFC offset value to ensure deterministic latency from one power cycle to another power cycle. In Figure 32 on page 102, the illegal *csr_lmfc_offset* values of 1, 2, and 3 causes lane de-skew error because the RBD buffer size has exceeded.

Figure 32. Selecting Illegal LMFC Offset Value for RX, Causing Lane Deskew Error

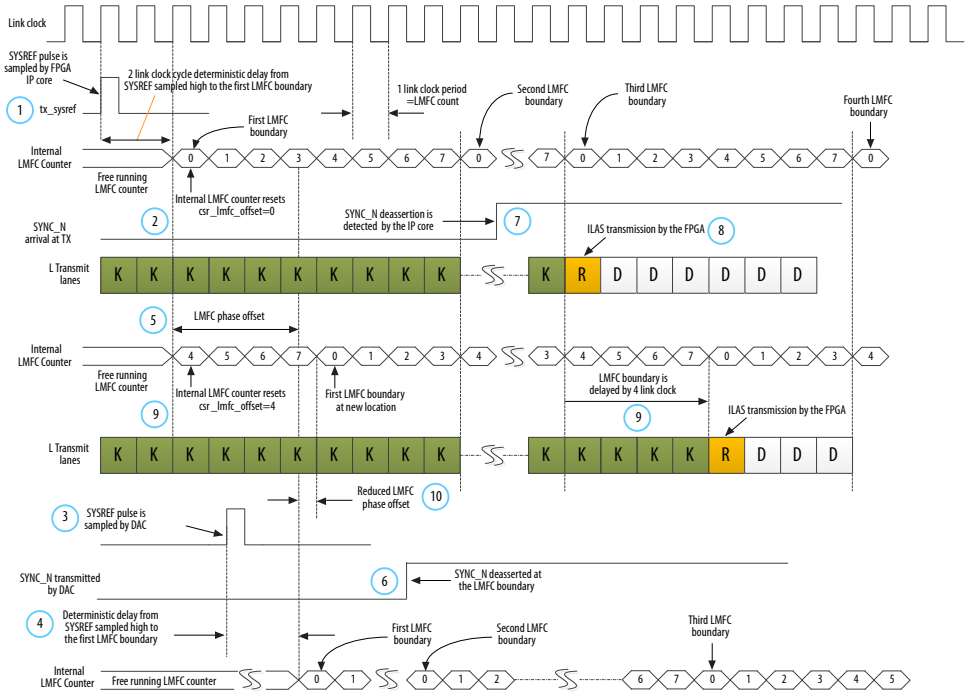


You can use the TX LMFC offset to align the LMFC counter in IP core to the LMFC counter in DAC.

Figure 33. Example of Reducing LMFC Phase Offset between TX and RX LMFC Counter

Sequence of events in the diagram:

1. SYSREF pulse arrives at the FPGA IP core port, `tx_sysref`.
2. The IP core's internal LMFC counter resets after two link clock cycles.
3. SYSREF pulse is sampled by the DAC.
4. The DAC's internal LMFC counter resets after a deterministic delay.
5. The LMFC phase offset is ~3.5 link clock cycles.
6. The DAC deasserts SYNC_N at the LMFC boundary.
7. SYNC_N deassertion is detected by the JESD204B IP core.
8. Because SYNC_N deassertion is detected after the second LMFC boundary at the FPGA, ILAS transmission begins at the third LMFC boundary.
9. The *csr_lmfc_offset* is set to 4. This delays the TX LMFC boundary by 4 link clock cycles. If *csr_lmfc_offset* is set to 5, the TX LMFC boundary is delayed by 3 link clock cycles.
10. The LMFC phase offset between the TX and RX LMFC reduces to 0.5 link clock cycle.



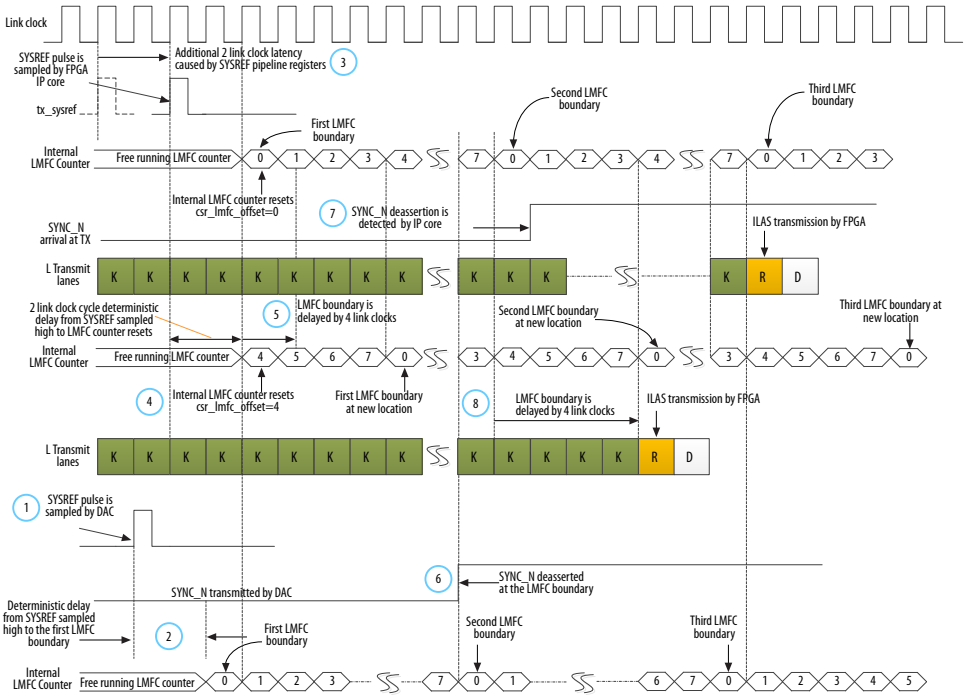
Alternative to tuning RBD offset at the DAC, adjusting TX LMFC offset in the FPGA helps you to achieve deterministic latency. You should perform multiple power cycles and read the RBD counts at the DAC to determine whether deterministic latency is achieved and RBD elastic buffer size has not exceeded.

The SYSREF pipeline registers in the FPGA introduce additional latency to SYSREF when detected by the IP core. Therefore, you can use TX LMFC offset to reduce or eliminate this additional latency. The next figure illustrates the technique of optimizing latency using TX LMFC offset.

Figure 34. Optimizing IP Core Latency Using TX LMFC Offset

Sequence of events in the diagram:

1. The DAC samples the SYSREF pulse.
2. The DAC's internal LMFC counter resets after a deterministic delay.
3. The SYSREF pipeline registers introduces an additional 2 link clock latency.
4. The *csr_lmfc_offset* field is set to 4. The IP core internal LMFC counter resets after 2 link clock cycles.
5. The LMFC boundary is delayed by 4 link clock.
6. The DAC deasserts SYNC_N at the LMFC boundary.
7. SYNC_N deassertion is detected by the JESD204B IP core.
8. Because LMFC boundary is delayed by 4 link clock, the IP core detects the SYNC_N deassertion before the second LMFC boundary. ILAS transmission begins at the second LMFC boundary instead of the third LMFC boundary (in Figure 33 on page 103). The latency is shortened by 4 LMFC counts or link clock cycles.

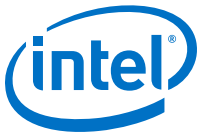


The *csr_lmfc_offset* field provides a convenient way to achieve deterministic latency and potentially optimizing the IP core latency. There are other ways that you can achieve deterministic latency by using the features available at the converters. Consult the converter manufacturer for details of these features.

5.4. Maintaining Deterministic Latency during Link Reinitialization

Link reinitialization occurs when the RX device deasserts the SYNC_N signal after link is established.

The converters resample the *SYSREF* signal and reset the internal LMFC counter. When the link is initially established, the IP core automatically clears the `csr_sysref_singledet` bit in the `syncn_sysref_ctrl` register (address 0x54)



when it detects the *SYSREF* pulse. The IP core does not automatically resample the *SYSREF* pulse unless the `jesd204_tx_avs_rst_n` or `jesd204_rx_avs_rst_n` signal is asserted.

If you are performing a link reset by asserting `txlink_rst_n` or `rxlink_rst_n` to reinitialize the link, set the `csr_sysref_singledet` bit to "1" to force the IP core to resample the *SYSREF* pulse without asserting the `jesd204_tx_avs_rst_n` or `jesd204_rx_avs_rst_n` signal.

6. JESD204B IP Core Debug Guidelines

This section lists some guidelines to assist you in debugging JESD204B link issues. Apart from applying general board level hardware troubleshooting technique like checking the power supply, external clock source, physical damage on components, a fundamental understanding of the JESD204B subsystem operation is important.

6.1. Clocking Scheme

To verifying the clocking scheme, follow these steps:

1. Check that the frame and link clock frequency settings are correct in the PLL Intel FPGA IP core or IOPLL Intel FPGA IP core.
2. Check the device clock frequency at the FPGA and converter.
3. For Subclass 1, check the SYSREF pulse frequency.
4. Check the management clock frequency. For the design examples using Arria V, Cyclone V, and Stratix V devices, this frequency is 100 MHz.

6.2. JESD204B Parameters

The parameters in both the FPGA and ADC should be set to the same values. For example, when you set K = 32 on the FPGA, set the converter's K value to 32 as well. Scrambling does not affect the link initialization in the CGS and ILAS phases but in the user data phase. When scrambling is enabled on the ADC, the FPGA descrambling option has to be turned on using the "Enable scramble (SCR)" option in the JESD204B IP core Platform Designer parameter editor. When scrambling is enabled on the FPGA, the DAC descrambling has to be turned on too.

Check these items:

- Turn off the scrambler and descrambler options as needed.
- Use single lane configuration and K = 32 value to isolate multiple lane alignment issue.
- Use Subclass 0 mode to isolate *SYSREF* related issues like setup or hold time and frequency of *SYSREF* pulse.

6.3. SPI Programming

The SPI interface configures the converter. Hence, it is important to check the SPI programming sequence and register bit settings for the converter. If you use the MIF to store the SPI register settings of the converter, mistakes may occur when modifying the MIF, for example, setting a certain bit to "1" instead of "0", missing or extra bits in a MIF content row.

Check these items:

- For example, in the ADI AD9250 converter, Intel recommends that you first perform register bit setting for the scramble (SCR) or lane (L) register at address 0x6E before setting the quick configuration register at address 0x5E.
- Determine that each row of the MIF has the same number of bits as the data width of the ROM that stores the MIF.

6.4. Converter and FPGA Operating Conditions

The transceiver channels at the converter and FPGA are bounded by minimum and maximum data rate requirements. Always check the most updated device data sheet for this info. For example, the Arria V GT device has a minimum data rate of 611 Mbps.

Ensure that the sampling rate of the converter is within the minimum and maximum requirements. For example, the ADC AD9250 has a minimum sampling rate of 40 Msps. For L = 2, M = 1 configuration, the minimum data rate of this ADC is calculated this way:

$$Minimum\ AD9250\ data\ rate = \frac{40 * 1 * 16 * \frac{10}{8}}{2} = 400\ Mbps$$

The minimum data rate for the JESD204B link is effectively 611 Mbps.

Check these items:

- Reduce the data rate or sampling clock frequency if your targeted operating requirement does not work.
- Verify the minimum and maximum data rate requirements in the device manufacturer's data sheet.

6.5. Signal Polarity and FPGA Pin Assignment

Verify that the transceiver channel pin assignments—SYNC_N and *SYSREF* (for Subclass 1 only)—device clock, and SPI interface are correct. Also verify the signal polarity of the differential pairs like SYNC_N and transceiver channels are correct.

Check these items:

- Review the schematic and board layout file to determine the polarity of the physical pin connection.
- Use assignment editor and pin planner to check the pin assignment and I/O standard for each pin.
- Use RTL viewer in the Intel Quartus Prime software to verify that the top level port are connected to the lower level module that you instantiate.

6.6. Creating a Signal Tap Debug File to Match Your Design Hierarchy

The Signal Tap and system console are very useful tools in debugging the JESD204B link related issues. The Signal Tap provides a dynamic view of signals.

For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, the Intel Quartus Prime software generates two files, `build_stp.tcl` and `<ip_core_name>.xml`. You can use these files to generate a Signal Tap file with probe points matching your design hierarchy.

The Intel Quartus Prime software stores these files in the *<debug stp directory>*. The *<debug stp directory>* is defined based on JESD204B wrapper and data path:

Table 26. File Directory

JESD204B Wrapper	Data Path	Debug stp directory
Both Base and PHY	Transmitter/ Duplex	*<ip_variant_name>*/ altera_jesd204_tx_mlpcs_*<Quartus_version>*/synth/debug/stp
	Receiver	*<ip_variant_name>*/ altera_jesd204_rx_mlpcs_*<Quartus_version>*/synth/debug/stp
Base only	Transmitter	*<ip_variant_name>*/altera_jesd204_tx_*<Quartus_version>*/synth/ debug/stp
	Receiver	*<ip_variant_name>*/altera_jesd204_rx_*<Quartus_version>*/synth/ debug/stp

Synthesize your design by running Analysis and Synthesis in the Intel Quartus Prime software.

1. Run analysis and synthesis.
2. Then open the Tcl console by clicking **View ➤ Utility Windows ➤ Tcl Console**.
3. Type the following command in the Tcl console:
   `source <debug stp directory>/build_stp.tcl`
4. To generate the STP file, type the following command:
   `main -stp_file <output stp file name>.stp -xml_file <input xml_file name>.xml -mode build`
5. The software generation script may not assign the Signal Tap acquisition clock in *<output stp file name>*`.stp`. Consequently, the Intel Quartus Prime software automatically creates a clock pin (`auto_stp_external_clock`) for each instance. To assign an acquisition clock for the generated STP file, Intel recommends that you perform the following assignments:

   JESD204B Duplex & Simplex (Both Base & PHY) or (PHY only) IP core:-

   - For rx_link instance, assign the `rxlink_clk` signal
   - For tx_link instance, assign the `txlink_clk` signal
   - For rx_phy instance, assign the input clock of the transceiver reset controller
   - For tx_phy instance, assign the input clock of the transceiver reset controller

   JESD204B Simplex (Base only) IP core:-

   - For rx_link instance, assign the `rxlink_clk` signal
   - For tx_link instance, assign the `txlink_clk` signal

Note: The GUI allows you to choose the appropriate instance for each IP core name if your design contains more than one JESD204B instances. For simplex core, you need to choose the RX instance followed by TX instance to generate the proper STP file.

6. Click **Save** to save the modified STP. A dialog box pops up with a message "`Do you want to enable Signal Tap File "<stp file name>" for the current project?`". Click **Yes**. Then, compile your design.
7. To program the FPGA, click **Tools ➤ Programmer**.
8. Open the generated STP file again if it has closed after step 6.
9. To observe the state of your IP core, click **Run Analysis** in the Signal Tap Logic Analyzer.

   You may see signals or Signal Tap instances that are red, indicating they are not available in your design. In most cases, you can safely ignore these signals and instances. They are present because the software generates wider buses and certain instances that your design does not include.

6.7. Debugging JESD204B Link Using System Console

The system console provides access to the JESD204B IP core register sets through the Avalon-MM interfaces.

To use the system console, your design must contain a Platform Designer subsystem with the JTAG-to-Avalon-MM Master bridge or Nios II Processor component. Connect the JESD204B IP core Avalon-MM interface to the Avalon-MM master through the Platform Designer interconnect directly if the IP core resides in the Platform Designer subsystem. Otherwise, connect the Avalon-MM interface through the Merlin slave translator if the IP core is not part of the Platform Designer subsystem.

PHY Layer

Verify the RX PHY status through these signals in the <*ip_variant_name*>**.v**:

- rx_is_lockedtodata
- rx_analogreset
- rx_digitalreset
- rx_cal_busy

Verify the TX PHY status through these signals in the <*ip_variant_name*>**.v**:

- pll_locked
- pll_powerdown
- tx_analogreset
- tx_digitalreset
- tx_cal_busy

Verify the RX_TX PHY status through these signals in the <*ip_variant_name*>**.v**:

- rx_is_lockedtodata
- rx_analogreset
- rx_digitalreset

- rx_cal_busy
- rx_seriallpbken
- pll_locked
- pll_powerdown
- tx_analogreset
- tx_digitalreset
- tx_cal_busy

Use the `rxphy_clk[0]` or `txphy_clk[0]` signal as sampling clock for the Signal Tap II.

For a normal operation of the JESD204B RX path, the `rx_is_lockedtodata` bit for each lane should be "1" while the `rx_cal_busy`, `rx_analogreset`, and `rx_digitalreset` bit for each lane should be "0".

For a normal operation of the JESD204B TX path, the `pll_locked` bit for each lane should be "1" while the `tx_cal_busy`, `pll_powerdown`, `tx_analogreset`, and `tx_digitalreset` bit for each lane should be "0".

Measure the `rxphy_clk` or `txphy_clk` frequency by connecting the clock to the *CLKOUT* pin on the FPGA. The frequency should be the same as link clock frequency for PCS option in Hard PCS or Soft PCS mode. The frequency is half of the link clock frequency for PCS option in PMA Direct mode.

Link Layer

Verify the RX PHY-link layer interface operation through these signals in the *<ip_variant_name>***_inst_phy.v**:

- jesd204_rx_pcs_data
- jesd204_rx_pcs_data_valid
- jesd204_rx_pcs_kchar_data
- jesd204_rx_pcs_errdetect
- jesd204_rx_pcs_disperr

Verify the RX link layer operation through these signals in the *<ip_variant_name>***.v**:

- jesd204_rx_avs_rst_n
- rxlink_rst_n_reset_n
- rx_sysref (for Subclass 1 only)
- rx_dev_sync_n
- jesd204_rx_int
- alldev_lane_aligned
- dev_lane_aligned
- rx_somf

Use the `rxlink_clk` signal as the sampling clock.

Verify the TX PHY-link layer interface operation through these signals in the *<ip_variant_name>***_inst_phy.v**:

- jesd204_tx_pcs_data
- jesd204_tx_pcs_kchar_data

Verify the TX link layer operation through these signals in the *<ip_variant_name>***.v**:

- jesd204_tx_avs_rst_n
- txlink_rst_n_reset_n
- tx_sysref (for Subclass 1 only)
- sync_n
- tx_dev_sync_n
- mdev_sync_n
- jesd204_tx_int

Intel recommends that you verify the JESD204B functionality by accessing the DAC SPI registers or any debug feature provided by the DAC manufacturer.

Figure 35. JESD204B Link Initialization

This is a Signal Tap image during the JESD204B link initialization. The JESD204B link has two transceiver channels (L = 2).

Description of the timing diagram:

- a. The JESD204B link is out of reset.
- b. The RX CDR is locked and PCS outputs valid characters to link layer.
- c. No running disparity error and 8b/10b block within PCS successfully decodes the incoming characters.
- d. The ADC transmits /K/ character or BC hexadecimal number to the FPGA, which starts the CGS phase.
- e. Upon receiving 4 consecutive /K/ characters, the link layer deasserts the `rx_dev_sync_n` signal.
- f. The JESD204B link transition from CGS to ILAS phase when ADC transmit /R/ or 1C hexadecimal after /K/ character.
- g. Start of 2nd multiframe in ILAS phase. 2nd multiframe contains the JESD204B link configuration data.
- h. Start of 3rd multiframe.
- i. Start of 4th multiframe.
- j. Device lanes alignment is achieved. In this example, there is only one device, the `dev_lane_aligned` connects to `alldev_lane_aligned` and both signals are asserted together.
- k. Start of user data phase where user data is streamed through the JESD204B link

Transport Layer

Verify the RX transport layer operation using these signals in the **altera_jesd204_transport_rx_top.sv**:

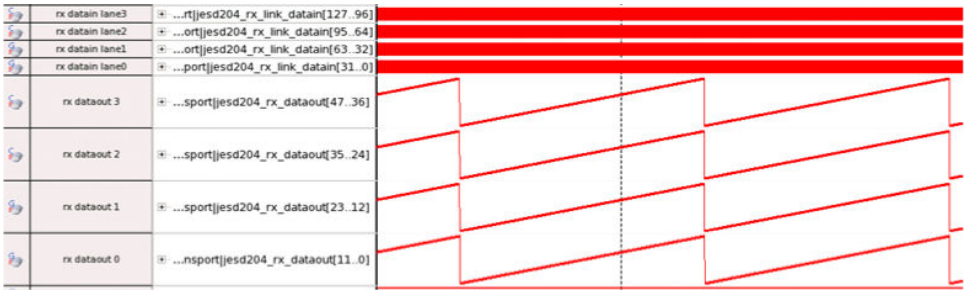
- jesd204_rx_dataout
- jesd204_rx_data_valid
- jesd204_rx_data_ready
- jesd204_rx_link_data_ready
- jesd204_rx_link_error
- rxframe_rst_n

Use the `rxframe_clk` signal as the sampling clock.

For normal operation, the `jesd204_rx_data_valid`, `jesd204_rx_data_ready`, and `jesd204_rx_link_data_ready` signals should be asserted while the `jesd204_rx_link_error` should be deasserted. You can view the ramp or sine wave test pattern on the `jesd204_rx_dataout` bus.

Figure 36. Ramp Pattern on the `jesd204_rx_dataout` Bus

This is a Signal Tap II image during the JESD204B user data phase with ramp pattern transmitted from the ADC.



Verify the TX transport layer operation using these signals in the **altera_jesd204_transport_tx_top.sv**:

- txframe_rst_n
- jesd204_tx_datain
- jesd204_tx_data_valid
- jesd204_tx_data_ready
- jesd204_tx_link_early_ready
- jesd204_tx_link_data_valid
- jesd204_tx_link_error

Use the `txframe_clk` signal as the sampling clock.

For normal operation, the `jesd204_tx_data_valid`, `jesd204_tx_data_ready`, `jesd204_tx_link_early_ready`, and `jesd204_tx_link_data_valid` signals should be asserted while the `jesd204_tx_link_error` should be deasserted. You

can verify the user data arrangement (shown in the data mapping tables in the TX Path Data Remapping section in *Design Examples for JESD204B IP Core User Guide*) by referring to the `jesd204_tx_datain` bus.

Related Information

- V-Series Transceiver PHY User Guide
- Intel Arria 10 Transceiver PHY User Guide
- Intel Cyclone 10 GX Transceiver PHY User Guide
- Intel Stratix 10 E-Tile Transceiver PHY User Guide
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
- Intel Stratix 10 GX 2800 L-Tile ES-1 Transceiver PHY User Guide
- AN 696: Using the JESD204B MegaCore Function in Arria V Devices
  More information about the performance and interoperability of the JESD204B IP core.
- AN 729: Implementing JESD204B IP Core System Reference Design with Nios II Processor As Control Unit
  An example of implementing a full-featured software control flow with various user commands in a JESD204B system that incorporates a Nios II processor
- JESD204B Reference Design
  Available design examples in Intel FPGA Design Store.
- Design Examples for JESD204B IP Core User Guide
  More information about the TX path data remapping in the Transport Layer.
- JESD204B Intel Arria 10 FPGA IP Design Example User Guide
- JESD204B Intel Stratix 10 FPGA IP Design Example User Guide
- JESD204B Intel Cyclone 10 FPGA IP Design Example User Guide

7. JESD204B Intel FPGA IP Document Archives

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

IP Core Version	User Guide
17.1	JESD204B IP Core User Guide
17.0	JESD204B IP Core User Guide
16.1	JESD204B IP Core User Guide
16.0	JESD204B IP Core User Guide

8. Document Revision History for the JESD204B Intel FPGA IP User Guide

Document Version	Intel Quartus Prime Version	Changes
2018.05.07	18.0	• Renamed JESD204B IP core to JESD204B Intel FPGA IP as per Intel rebranding. • Added support for Intel Cyclone 10 GX devices. • Added simulation setup and run scripts for the Cadence Xcelium Parallel simulator. • Added links to the *JESD204B Intel FPGA IP Design Example for Intel Cyclone 10 GX Devices User Guide*. • Edited a typo in in the *Brief Information About the JESD204B IP Core* table. Changed Platform Designer (Standard) to Platform Designer. • Revised the resource utilization data and speed grade information for version 18.0. • Updated the *JESD204B IP Core Parameters* and *Signals* sections with Intel Cyclone 10 GX information. • Edited the steps in the *Creating a Signal Tap Debug File to Match Your Design Hierarchy* section. • Added a note in the *Testbench Simulation Flow* section that for Intel Stratix 10 devices, reset deassertion staggering of TX/RX analog and digital reset happens before the assertion of TX/RX ready.

Date	Version	Changes
November 2017	2017.11.06	• Updated `alldev_lane_aligned` signal description. • Updated bonded channel requirements for Intel Stratix 10 and Intel Arria 10 devices. • Updated instances of Qsys to Platform Designer. • Updated steps to simulate the testbench design using the Aldec Riviera-PRO simulator. • Removed note stating that the dynamic reconfiguration is not supported for the JESD204B IP core testbench. • Added Base only, or Simplex TX configuration in *Preset Configurations for JESD204B IP Core Testbench* table. • Added Provide Separate Reconfiguration Interface for Each Channel parameter in the *JESD204B IP Core Parameters* table. • Updated Link Clock F_{MAX} (MHz) for all devices in the *JESD204B IP Core FPGA Performance* table. • Added supported data rate note to Intel Arria 10 variants in *JESD204B IP Core FPGA Performance* table. • Updated rx_phy and tx_phy assignments in *Creating a Signal Tap Debug File to Match Your Design Hierarchy*.

continued...

Date	Version	Changes
		• Updated the data rate support for Intel Stratix 10 devices: — For speed grade 1 to up to 16.0 Gbps. — For speed grade 2 to up to 13.5 Gbps. — For speed grade 3 to up to 12.5 Gbps. • Updated and added example timing diagrams in *Subclass 2 Operating Mode* subsection. • Updated transceiver interface signals for transmitter and receiver signals.
May 2017	2017.05.08	• Updated description for PLL/CDR Reference Clock Frequency in *JESD204B IP Core Parameters*. • Added `somf[]` for transmitter signal. • Updated *Run-Time Configuration* to include statement on JESD204B IP core parameterization for Intel Stratix 10 devices. • Added note to *Registers* to indicate that disabled run-time access for registers in Intel Stratix 10 devices. • Updated data path preset value to simplex TX and simplex RX in *Preset Configurations for JESD204B IP Core Testbench* table. • Clarified the device family support for Stratix 10 devices. • Added Transmitter and Receiver signal diagrams. • Added **Share Reconfiguration Interface** parameter in *JESD204B IP Core Parameters* table. • Added note and description to Share Reconfiguration Interface in *JESD204B IP Core Parameters*. • Added links to *Stratix 10 device Datasheet*, *Stratix 10 L-Tile Transceiver PHY User Guide*, and *Stratix 10 H-Tile Transceiver PHY User Guide*. • Added `tx_analogreset_stat`, `rx_analogreset_stat`, `tx_digitalreset_stat`, and `rx_digitalreset_stat` signals and descriptions. • Updated *ADC-FPGA Subsystem Reset Scheme* and *FPGA-DAC Subsystem Reset Scheme*.
October 2016	2016.10.31	• Updated *Example 2* in *Clock Correlation*. • Updated steps in *FPGA–DAC Subsystem Reset Sequence*. • Updated figure and title for *JESD204B Subsystem with Shared Transceiver Reference Clock and Core Clock* and *JESD204B Subsystem with Separate Transceiver Reference Clock and Core Clock*. • Updated *Subclass 1 Deterministic Latency and Support for Programmable Release Opportunity* figure. • Updated *Channel Bonding* description.
May 2016	2016.05.02	• Updated the IP Core release information. • Updated the data rate support—uncharacterized support for data rates of up to 15 Gbps. • Updated the data rate for Intel Arria 10 and Arria V GT/ST in the JESD204B IP Core FPGA Performance table. • Updated the JESD204B IP Core FPGA Resource Utilization table. • Updated the **PLL/CDR Reference Clock Frequency** parameter description. • Updated the preset values for **PLL/CDR Reference Clock Frequency**, **Link Clock**, and **AVS Clock** in Table 3-8: Preset Configurations for JESD204B IP Core Testbench. • Updated Figure 4-8 to illustrate shared clocking and renamed the title to "*JESD204B Subsystem with Shared Transceiver Reference Clock and Core Clock*". • Added a new figure to illustrate separate clocking—*Figure 4-9: JESD204B Subsystem with Separate Transceiver Reference Clock and Core Clock*.

continued...

Date	Version	Changes
		• Added new sections: — ADC–FPGA Subsystem Reset Sequence on page 71 — FPGA–DAC Subsystem Reset Sequence on page 73 — Creating a Signal Tap Debug File to Match Your Design Hierarchy on page 108 • Updated the TX path **CONTROL_BUS_ WIDTH** parameter description. • Revised the clock domain for `jesd204_tx_data_ready` signal to txframe_clk. • Updated the description for the following registers in the register map: — rx_regmap: • `csr_frame_data_ready_err` • `csr_pcfifo_full_err` • `csr_pcfifo_empty_err` — tx_regmap: • `csr_pcfifo_full_err` • `csr_pcfifo_empty_err` • Added links to archived document in JESD204B Intel FPGA IP Document Archives on page 115.
November 2015	2015.11.02	• Added data rate support of up to 13.5 Gbps for Intel Arria 10 and 7.5 Gbps for Arria V GT/ST devices. • Updated the IP core FPGA performance and resource utilization values. • Added a new table to define the clock network selection for bonded mode in channel bonding. • Added a new selection for **PCS Option** parameter—Enabled PMA Direct. • Updated the preset value for link clock in JESD204B IP Core Testbench on page 40. • Updated the formula and description for TX/RX PHY clock. • Updated the device clock section to recommend user to supply the device clock with the same frequency as the link clock. • Updated the description of `txlink_clk`, `txphy_clk[]`, and `rxphy_clk[]` signals. • Changed the default value for RX Phase Compensation FIFO empty error enable (`csr_pcfifo_empty_err_en`) CSR to 0. Refer to the *RX register map* for details. • Added a new section—*Design Example with Nios II Processor Control Unit*. • Added a new topic – Maintaining Deterministic Latency during Link Reinitialization on page 104. • Changed instances of *Quartus II* to *Quartus Prime*.
May 2015	2015.05.04	• Added support for Cyclone V FPGA device family. • Updated the JESD204B IP Core Configuration values: — M value from 1-32 to 1-256 — N' value from 4-32 to 1-32 • Updated the JESD204B IP Core FPGA Performance table. • Updated the JESD204B IP Core FPGA Resource Utilization table. • Added new parameters to the JESD204B IP Core Parameters table: — **Enable Capability Registers** — **Set user-defined IP identifier** — **Enable Control and Status Registers** — **Enable Prbs Soft Accumulators** — **Enable manual F configuration**

continued...

Date	Version	Changes
		• Added new topics: — Timing Constraints For Input Clocks on page 32 — JESD204B IP Core Deterministic Latency Implementation Guidelines on page 96 • Revised the note in "Simulating the IP Core Testbench" to state that VHDL is not supported in Aldec Riviera (for Intel Arria 10 devices only). • Updated the *Control Unit Process Flow* diagram.
December 2014	2014.12.15	• Updated the JESD204B IP Core FPGA Performance table with the data rate range. • Updated the JESD204B IP Core FPGA Resource Utilization table. • Updated the JESD204B IP Core Parameters table with the following changes: — Revised the parameter name of **Enable PLL/CDR Dynamic Reconfiguration** to **Enable Transceiver Dynamic Reconfiguration**. — Added information for a new parameter—**Enable Altera Debug Master Endpoint**. — Added details about the rule check for parameter N' value. • Added a new topic—Integrating the JESD204B IP Core in Platform Designer on page 30. • Updated *Overview of the JESD204B IP Core Block Diagram*, *Transmitter Data Path Block Diagram*, and *Receiver Data Path Block Diagram*. • Added a new table—Register Access Type Convention—to describe the access type for the IP core registers. • Added new signals description for `jesd204_tx_controlout` and `jesd204_rx_controlout`. • Added **CONTROL_BUS_WIDTH** parameter and description for the assembler and deassembler. • Added information on how to run the Tcl script using the Quartus II software before compiling the design example. • Updated the section on Debugging JESD204B Link Using System Console on page 109 with verification information for TX PHY-link layer interface, TX link layer, and TX transport layer operations.
June 2014	2014.06.30	• Updated Figure 2-1 to show a typical system application. • Updated the list of core key features. • Updated the Performance and Resource utilization values. • Updated the Getting Started chapter to reflect the new IP Catalog and parameter editor. • Added the following new sections to further describe the JESD204B IP core features: — Channel Bonding — Datapath Modes — IP Core Variation — JESD204B IP Core Testbench — JESD204B IP Core Design Considerations — TX Data Link Layer — TX PHY Layer — RX Data Link Layer — RX PHY Layer — Operation — Dynamic Reconfiguration — JESD204B IP Core Debug Guidelines • Updated the Clocking scheme section. • Added new transceiver signals that is supported in Intel Arria 10 devices.

continued...

Date	Version	Changes
		• Updated the Transport Layer section. • Added run-time reconfiguration parameter values in the System Parameters section. • Updated the file directory names.
November 2013	2013.11.04	Initial release.