HB0400

CoreJESD204BTX v3.1

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

50200400-6/5.19

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 6.0

Updated changes related to CoreJESD204BTX v3.1.

- Updated description for CF and HD parameters
- Added new section Clocking and Reset

1.2 Revision 5.0

Updated changes related to CoreJESD204BTX v3.0. SAR (83378): Add support for 8 lanes.

1.3 Revision 4.0

Updated changes related to CoreJESD204BTX v3.0.

1.4 Revision 3.0

Updated changes related to CoreJESD204BTX v2.3.

1.5 Revision 2.0

Updated changes related to CoreJESD204BTX v2.2.

1.6 Revision 1.0

Revision 1.0 was the first publication of this document. Created for CoreJESD204BTX v2.0.

Contents

Figures

Tables

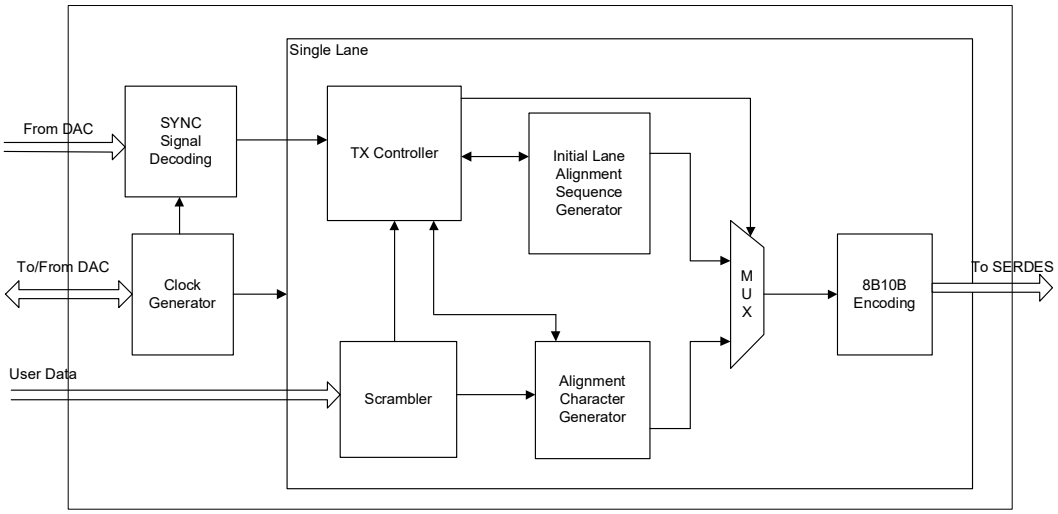
2 Introduction

2.1 Overview

CoreJESD204BTX is the transmitter interface of the JEDEC JESD204B standard. This specification describes a high speed serial interface for data converters.

Figure 1 shows the CoreJESD204BTX block diagram of a single lane.

Figure 1 • **CoreJESD204BTX Block Diagram**



2.2 JESD204BTX Blocks

CoreJESD204BTX consists of the blocks described below:

2.2.1 Alignment Character Generator

This block generates the alignment characters and frame position signals.

During a synchronization request this block generates the K28.5 (/K/) characters used during code group synchronization (CGS) state. The receiver device aligns its bit boundaries to these /K/ characters and when it detects four or more consecutive /K/ characters it will no longer request synchronization. After CGS this block monitors the frame position and asserts the SOF and SOMF outputs to indicate when the frame is at the start of a frame or multi-frame.

After the Initial Lane Alignment (ILA) sequence frame alignment is monitored via alignment characters. The Alignment Character Generator (ACG) replaces octets under certain conditions at the end for a frame with these alignment characters. The alignment character K28.7 (/F/) is used to mark end-of-frame and the alignment character K28.3 (/A/) is used to mark end-of-multi-frame. When the receiver device detects one of these alignment characters it reconstructs the octet back into its original value, therefore no extra bandwidth is required for frame alignment. Replacement characters are generated based on the configuration of scrambling as enabled (SCR=1) or disabled (SCR=0). When scrambling is disabled ACG replaces the user data with an alignment character when the last octet in the current frame equals the last octet in the previous frame, but only if character replacement did not occur in the previous frame or if the current frame is the end of a multi-frame. When scrambling is enabled ACG replaces the user data with the alignment character /F/ when the last scrambled octet in the current frame equals 0xFC but only if the character replacement did not occur in the previous frame and it is not the end of a multi-frame. If the last scrambled octet of the current frame equals 0x7C and is the end of a multi-frame the user data is replaced with the alignment character /A/ and will make this replacement even if character replacement occurred in the previous frame.

Note: Refer to section 5.3.3.4 of the JESD204B.01 specifications for more details.

2.2.2 ILA Sequence Generator

This block generates the initial lane alignment (ILA) sequence.

The first control symbol of every multi-frame in the ILA sequence is K28.0 (/R/) which indicates the start of the subsequence. The last control symbol of every frame in the ILA is /A/ which indicates the end of multi-frame. The receiver lane(s) align to the first /R/ symbol during initialization. The second frame of the multi-frame has the link configuration data. The control symbol K28.4 (/Q/) indicates that the next octet is the start of the link configuration data. The parameters in the link configuration data are used to indicate to the receiver device how the transmitter is configured. The parameter 'L' is set by lane number. For subclass 2 the adjustment information is calculated and send in PHADJ, ADJDIR and ADJCNT parameters of the link configuration data. Although the JESD204B.01 specification indicates that the checksum (FCHK) in the link configuration mapping is the sum of all the fields and not the sum of all the octets created by the mapping some data convertor manufactures have the checksum as the sum of all the octets. To cover both checksum types the parameter/generic FIELD_OCTET can select which type of checksum to calculate.

When the SYNC Signal Decoding block detects a synchronization request the Alignment Character Generator block will transmit a stream of consecutive K28.5 (/K/) characters until the synchronization request has been completed and the TX Controller is ready to send the ILA sequence. When the synchronization request has been completed the TX Controller will start the ILA sequence at the start of the next frame clock boundary for subclass 0 and at the start of the next LMFC boundary for subclasses 1&2. This control over the ILA sequence is used for lane alignment and the receiver device uses this information to align its lanes. Since subclasses 1&2 start the ILA sequence at the start of the next LMFC boundary only the four multi-frames in the ILA sequence are required to align the receiver device lanes. However, since Subclass 0 starts the ILA sequence at the start of the next frame clock boundary additional alignment frames may be required for lane alignment. The parameter/generic ILA_MFS indicates the number of multi-frames in the ILA sequence (default is 4). If any addition, alignment frames are required this parameter can be configured up to 256. The additional alignment frames will have control characters K28.7 (/F/) at the end of each frame and K28.3 (/A/) control characters at the end of each multi-frame.

Note: Refer to 5.3.3.5, 8.2 and 8.3 of the JESD204B.01 specifications for details.

2.2.3 SYNC Signal Decoding

This block controls the sync request by decoding the SYNC_N signal.

At reset or power-on the transmitter goes through initialization and synchronization request is sent to the TX Controller. TX Controller will instruct the Alignment Character Generator to transmit K28.5 (/K/) characters during a synchronization request. When SYNC_N de-asserts the synchronization request is complete, but transmission of /K/ characters will continue to the start of the next frame clock boundary for subclass 0 and to the start of the next LMFC boundary for subclasses: 1 and 2. If SYNC_N is asserted for another 4 consecutive rising edge of the frame clock a re-initialization request is triggered and the same process as above is executed.

Note: Refer to section 8.4 of the JESD204B.01 specifications for details on SYNC signal decoding.

2.2.4 Clock Generator

This block generates the internal LMFC and frame clock. When subclass 1 is selected it also generates SYSREF_OUT output and monitors SYSREF_IN input for deterministic latency.

Note: More information can be found in the JESD204B.01 specifications. See Annexure G for clock generation information and section 6 for deterministic latency.

2.2.5 8B10B Encoder

This block encodes data using 8B10B encoding techniques. The ENCODER_EN parameter is used to enable or disable the 8B10B encoder, fourth generation families require it to be enabled while PolarFire requires it to be disabled since this family of devices has a hard 8B10B encoder in the transceiver block.

The 8B10B encoder is a marriage of two sub-blocks, the 5b6b and the 3b4b encoders. It provides DC-balance, bounded disparity and clock recovery for the receiver. The purpose of the encoders is to convert 8-bit data into a 10-bit code. The 10-bit code contains an equal number of 0's and 1's. In addition, the code is built so that no more than five consecutive 0's or 1's is ever transmitted, if the parameter/generic SERDES_MODE is set to 1 each lane has a 20-bit output, which will have two 10-bit codes. If SERDES_MODE is set to 2 each lane has a 10-bit output, which will have one 10-bit.

2.2.6 TX Controller

This block controls the triggering of the ILA sequence, generation of alignment characters, and the data presented to the 8B10B encoder.

During a synchronization request this block will control the Alignment Character Generator (ACG) and it will transmit a stream of continuous K28.5 /K/ characters. The ACG data path is selected on the MUX and goes through to the 8B10B Encoder. The minimum generation of /K/ symbols is 1 frame and 9 octets. Once the receiver device aligns its bit boundaries to these /K/ characters the synchronization request will de-assert (that is, SYNC_N goes HIGH). Then this block will control the ILA Sequence Generator and trigger the ILA sequence at the start of the next frame clock boundary for subclass 0 and to the start of the next LMFC boundary for subclasses 1 and 2. The ILA Sequence Generator data path is selected on the MUX and goes through to the 8B10B Encoder. When the ILA sequence has complete the ILA Sequence Generator informs this block and now user data can be transmitted so the data path changes the back to the ACG data.

This block also controls the character replacement. The behavior of the character replacement depends on if scrambling is enabled or disabled and is described below.

When scrambling is disabled, when the last octet in the current frame, not coinciding with the end of a multi-frame, equals the last octet in the previous frame, the transmitter will replace the current last octet and encode it as control character K28.7 (/F/). However, if an alignment character was already transmitted in the previous frame, the original octet will be encoded. When the last octet in the current frame at the end of a multi-frame equals the last octet in the previous frame, the transmitter will replace the current last octet and encode it as control character K28.3 (/A/), even if a control character was already transmitted in the previous frame.

When scrambling is enabled, when the last scrambled octet in a frame, but not at the end of a multi-frame, equals 0xFC, the transmitter will encode it as a control character /F/. When the last scrambled octet in a multi-frame equals 0x7C, the transmitter will encode it as a control character /A/.

Note: Refer to section 8.1 of the JESD204B.01 specifications for details.

2.2.7 Scrambler

This block scrambles data based on the polynomial 1 + x14 + x15.

Scrambling is enabled by setting parameter/generic SCR to 1. Scrambling must only be enabled if the receiver has descrambling enabled or the receiver can enable scrambling based on the link configuration parameters in the ILA sequence. Scrambling of data is used to avoid electromagnetic compatibility and interface problems in sensitive application. Spectral peaks are produced when the same data pattern is repeated over a period of time. This polynomial offers a period of 32,767 which is long enough to meet the spectral requirements of sensitive application. The scrambler is self-synchronous and also allows the descrambler in the receiver to synchronize within two octets. The reason scrambling is optional is because the digital operations of the interface cause switching noise which makes scrambling a disadvantage for some applications. Switching noise causes current spikes which cause sensitive analogue portions of the design to degrade or even malfunction.

More information can be found in the JESD204B.01 specifications. Refer to section 5.2 for information about scrambling and view Figure D.4 for the scrambler implementation.

2.3 Features

CoreJESD204BTX implements the transmitter interface of the JESD204B standard and has the following features:

- Supports 1 to 8 lanes
- Performs user-enabled scrambling
- Generates an initial lane alignment sequence
- Performs alignment character generation
- Sources link configuration data with user selected parameter values during initial lane synchronization sequence
- Performs the alignment character generation
- Performs user-enabled 8B10B encoding
- Supports Subclasses 0, 1, and 2
- Internal clock generation
- Sync~ signal decoding
- Supports data width of 16, 32, or 64 bits

Note: Octet to Frame Stream conversion is not performed by the core.

2.4 Core Version

This handbook is for CoreJESD204BTX version 3.1.

2.5 Supported Families

- PolarFire®
- RTG4™
- SmartFusion®2
- IGLOO®2

2.6 Device Utilization and Performance

CoreJESD204BTX has been implemented in the following Microsemi device families. A summary of the implementation data for CoreJESD204BTX is provided in Table 1.

Table 1 • **CoreJESD204BTX Utilization for Data Width 16 With Encoder**

	Logic Elements			Utilization		
Family	**Sequential**	**Combinatorial**	**Total**	**Device**	**Total %**	**Performance (MHz)**
PolarFire	2421	2576	4997	PA5M500	0.52	240
RTG4	2675	2800	5475	RT4G150	1.80	170
IGLOO2	2516	2674	5190	M2GL150T	1.78	200
SmartFusion2	2501	2640	5141	M2S150T	1.76	200

Note: Data in this table were achieved using synthesis and layout settings optimized from speed. The following parameters/generics were set: L=3, F=2, K=9, ENCODER_EN = 1 & D_WIDTH=16.

Table 2 • **CoreJESD204BTX Utilization for Data Width 32 With Encoder**

Family	Logic Elements			Utilization		Performance (MHz)
	Sequential	Combinatorial	Total	Device	Total %	
PolarFire	4591	4746	9337	PA5M500	0.97	215
RTG4	4881	5135	10016	RT4G150	3.30	150
IGLOO2	4627	4818	9445	M2GL150T	3.23	170
SmartFusion2	4477	4756	9233	M2S150T	3.16	170

Note: Data in this table were achieved using synthesis and layout settings optimized from speed. The following parameters/generics were set: L=3, F=2, K=9, ENCODER_EN = 1 & D_WIDTH=32.

Table 3 • **CoreJESD204BTX Utilization for Data Width 64 With Encoder**

Family	Logic Elements			Utilization		Performance (MHz)
	Sequential	Combinatorial	Total	Device	Total %	
PolarFire	8887	9396	18283	PA5M500	1.90	180
RTG4	9176	9708	18884	RT4G150	6.22	110
IGLOO2	9000	9566	18566	M2GL150T	6.35	130
SmartFusion2	9000	9566	18566	M2S150T	6.35	130

Note: Data in this table were achieved using synthesis and layout settings optimized from speed. The following parameters/generics were set: L=3, F=2, K=9, ENCODER_EN = 1 & D_WIDTH=64.

Table 4 • **CoreJESD204BTX Utilization for Data Width 16 Without Encoder**

Family	Logic Elements			Utilization		Performance (MHz)
	Sequential	Combinatorial	Total	Device	Total %	
PolarFire	1575	1549	3124	PA5M500	0.32	240

Note: Data in this table were achieved using synthesis and layout settings optimized from speed. The following parameters/generics were set: L=3, F=2, K=9, ENCODER_EN = 0 & D_WIDTH=16.

Table 5 • **CoreJESD204BTX Utilization for Data Width 32 Without Encoder**

Family	Logic Elements			Utilization		Performance (MHz)
	Sequential	Combinatorial	Total	Device	Total %	
PolarFire	2835	2700	5535	PA5M500	0.58	218

Note: Data in this table were achieved using synthesis and layout settings optimized from speed. The following parameters/generics were set: L=3, F=2, K=9, ENCODER_EN = 0 & D_WIDTH=32.

Table 6 • **CoreJESD204BTX Utilization for Data Width 64 Without Encoder**

Family	Logic Elements			Utilization		Performance (MHz)
	Sequential	Combinatorial	Total	Device	Total %	
PolarFire	4934	1502	6436	PA5M500	0.67	197

Note: Data in this table were achieved using synthesis and layout settings optimized from speed. The following parameters/generics were set: L=3, F=2, K=9, ENCODER_EN = 0 & D_WIDTH=64.

3 Functional Description

CoreJESD204BTX, shown in Figure 2, consists of optional scrambling, ILA sequence/alignment character generation, SYNC_N signal decoding, optimal 8B10B encoding, and clock generation.

3.1 Data Link Layer

CoreJESD204BTX implements the JEDEC JESD204B transmitter interface.

Figure 2 • **Data Link Layer**

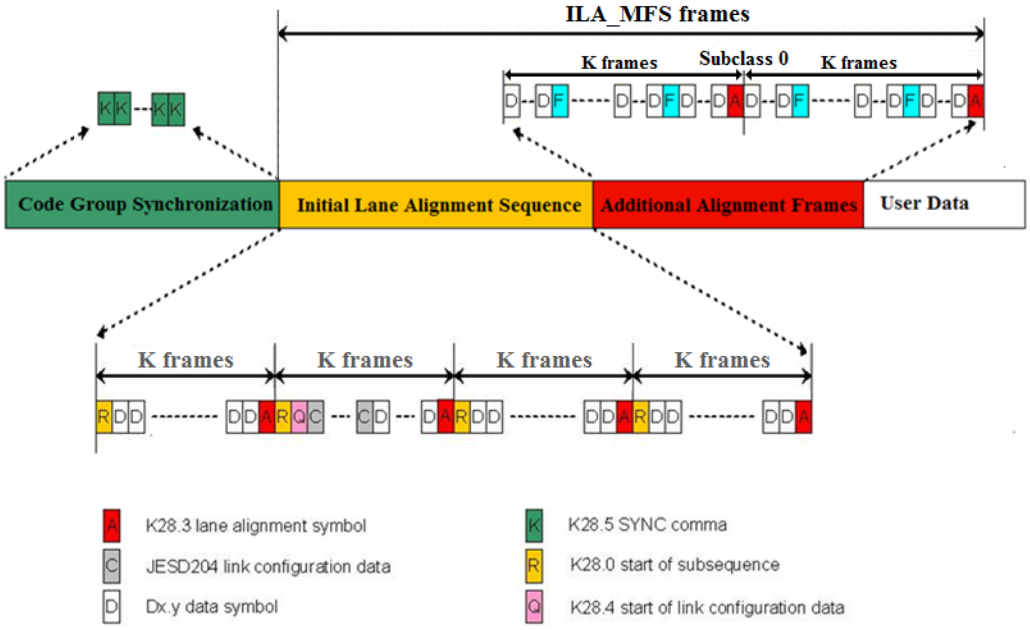


Figure 2 shows the data link layer of CoreJESD204BTX. Code Group Synchronization (CGS) is triggered when the receiver issues a synchronization request via the synchronization interface. During a synchronization request the transmitter emits a continuous stream of K28.5 (/K/) control symbols. The receive will then align its bit boundaries to this symbol and will deactivate the synchronization request when it receives at least four successive /K/ symbols. Then the Initial Lane Alignment (ILA) sequence is transmitted. The first control symbol of every multi-frame in the ILA sequence is K28.0 (/R/) which indicates start of the subsequence.

The last control symbol of every frame in the ILA is K28.3 (/A/) which indicates the end of multi-frame. The receiver lane(s) align to the symbol during initialization. The second frame of the multi-frame has the link configuration data. The control symbol K28.4 (/Q/) indicates that the next octet is the start of the link configuration data. The receiver compares its parameters to the parameters received in the link configuration data or configure it to these values. Subclass 0 support addition, alignment frames, if more frames are required for alignment. The additional alignment frames will have K28.7 (/F/) symbols at the end of the frame and /A/ symbols at the end of multi-frame. After alignment completes the user data can be transmitted.

3.2 JESD204B Sub-classes

CoreJESD204BTX supports sub-class 0, 1, and 2. A summary of each subclass is given below.

3.2.1 Sub-class 0

Sub-class 0 allows the device to be backward compatible with the JESD204A standard. It offers no support for deterministic latency.

3.2.2 Sub-class 1

Subclass 1 offers support for deterministic latency using SYSREF signaling.

During a synchronization request SYSREF_OUT will only assert at the start of a LMFC period and when SYNC_N is asserted. The assertion of SYSREF_OUT indicates to the receiver when the start of a LMFC period occurs. The receiver has the option to shift the phase of its internal clocks based on SYSREF_OUT assertion. The minimum duration SYSREF_OUT will assert for is a quarter of a LMFC period.

The assertion of SYSREF_IN input indicates the start of a LMFC period in the transmitter. During a synchronization request the receiver asserts SYNC_N and when SYSREF_IN asserts it will cause LMFC and frame clock to phase shift to sync to the receivers LMFC period for deterministic latency support. When SYNC_N is de-asserted any assertion of SYSREF_IN will be ignored so that the LMFC and frame clock positions are locked.

3.2.3 Sub-class 2

Sub-class 2 is similar to sub-class 1 except it offers support for deterministic latency using SYNC~ sampling.

The de-assertion of SYNC_N is synchronous to the internal frame clock and will only occur at the LMFC boundary. This control means that the LMFC phase information is sent to the transmitter device through the SYNC interface. The de-asserts of SYNC_N relative to its LMFC and the correction information shall be contained in PHADJ, ADJDIR and ADJCNT parameters on the link configuration data. The adjustment resolution is set to 1 clock tick (that is, two octets) in the transmitter and is stored in ADJCNT. When PHADJ is set (= 1) a phase adjust has been requested by the transmitter. ADJDIR value indicates an advance (= 0) or delay (= 1) adjustment. Since The receiver device has the option to make adjustments based on the adjustment information received in the link configuration data. If the receiver is a logic device, there are no requirements to adjust the LMFC phase, but if the receiver is a convertor device, it shall be able to adjust its LMFC phase when PHADJ is set (=1) and will make the adjustments based of the information in ADJDIR and ADJCNT.

Note: More information about deterministic latency can be found in section 6 of the JESD204B.01 specifications.

3.3 SERDES Interface

For RTG4/IGLOO2/SmartFusion2 family devices CoreJESD204BTX will be connected to the transmitter lanes of the SERDES interface macro. The SERDES interface must be configured in EPCS mode as this macro will be used as a standalone SERDES. EPCS mode has two default data rates: 1.25 Gbps and 2.5 Gbps per lane. This means 10 Gbps can easily be achieved across the 4 lanes of CoreJESD204BTX. The SERDES interface can be reconfigured through the APB interface to allow EPCS mode to achieve higher data rates.

For PolarFire family devices the SERDES has a hardened 8B10B encoder which means CoreJESD204BTX can be configured with its encoder removed (that is, ENCODER_EN = 0). CoreJESD204BTX will connect to the 8B10B interface on the SERDES. The PolarFire SERDES can be configured up to 64bit data width (8 octets) and CoreJESD204BTX supports this data width which means up to 12.5Gbps data rate is achievable per lane.

4 Interface

4.1 Configuration Parameters

CoreJESD204BTX has parameters (Verilog) or generics (VHDL) for configuring the RTL code, described in Table 7. All parameters and generics are integer types.

Table 7 • **CoreJESD204BTX Parameters and Generics Descriptions**

Name	Range	Default Value	Description
CF	0 to 7	0	Number of control words per frame clock period per link. It controls which lanes will carry control words. CF=0 means that no control words are used. Other allowed CF values are common sub-divisors of Number of lanes (L) and Number of converters (M). The L lanes are divided into CF groups of L/CF lanes. Each group of lanes transmits the samples of M/CF converters. After these samples a control word is inserted, containing in successive order the control bits belonging to these samples. If the control word fits on a single lane, it is not allowed to break it over a lane boundary.
CS	0 to 3	0	Number of control bits per sample
F	2 to 256	2	Number of octets per frame
HD	0 to 1	0	High density format. HD controls whether a sample may be divided over more lanes. In the Low Density mode (HD=0), partial conversion words at the end of a group of F octets are avoided by adding more tail bits (TT) after the last full nibble group (NG) in the group if necessary. In the High Density mode (HD=1), the conversion words may break at the lane boundary.
JESDV	0, 1	1	JESD204 version
			0: JESD204A
			1: JESD204B
K	4 to 32	9	Number of frames per multi-frame
L	0 to 7	0	Number of lanes per converter device (link) **Note:** L+1= No. of lanes
M	1 to 256	1	Number of converters per device
N	1 to 32	16	Converter resolution
N’	1 to 32	16	Total number of bits per sample
S	1 to 32	2	Number of samples per converter per frame cycle
SCR	0 to 1	1	Scrambling enabled
SUBCLASSV	0 to 2	0	Device subclass version
			0: Subclass 0
			1: Subclass 1
			2: Subclass 2
ILA_MFS	4 to 256	4	Initial lane alignment, multi-frame size. Minimum size is 4; however, more frames may be needed for the receive to align its lane(s).

Table 7 • **CoreJESD204BTX Parameters and Generics Descriptions**

FIELD_OCTET	0 to 1	1	Configuration Options parameter which selects how the FCHK (checksum) of the link configuration parameters will be calculated 0: Field addition 1: Octet addition
SERDES_MODE	1 to 2	1	Configuration Options parameter which selects the width of the input data 1 : ((D_WIDTH/8) * 10) 2 : 10bit (when ENCODER_EN = 1), : 8bit (when ENCODER_EN = 0) **Note:** When SERDES_MODE=2 LANE_CLK is a slower frequency than EPCS Tx Clock.
D_WIDTH	16, 32, or 64	16	Configuration Options parameter which selects the user data width. 0 : 16bit (2 octet) 1 : 32bit (4 octet) 2 : 64bit (8 octet)
ENCODER_EN	0 to 1	1	Configuration Options parameter which includes or removes the encoder from the core. 0 : Removed 1 : Included

Note: A multiframe is defined as a group of K successive frames, which are made up of F successive octets, where K is valid from 4 to 32 and F is valid from 2 and 256, such that the number of octets per multiframe is between 18 and 1024.

4.2 I/O Signals

The port signals for the CoreJESD204BTX macro are shown in Figure 3 and Figure 4.

Figure 3 • **CoreJESD204BTX I/O Signal Diagram (4 lanes) with Encoder**

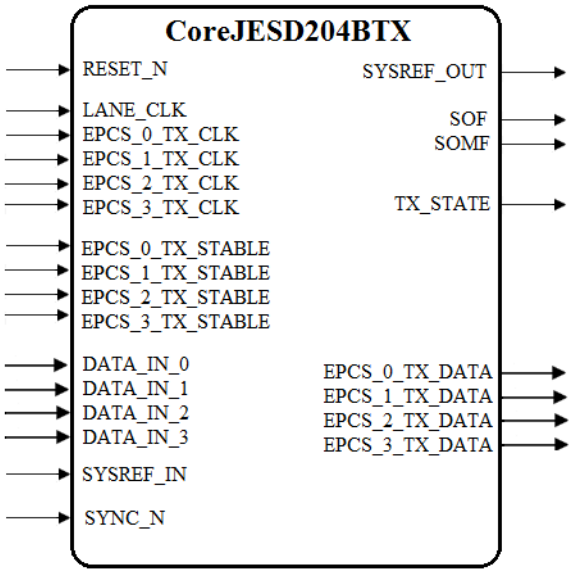


Figure 4 • **CoreJESD204BTX I/O Signal Diagram (4 lanes) without Encoder**

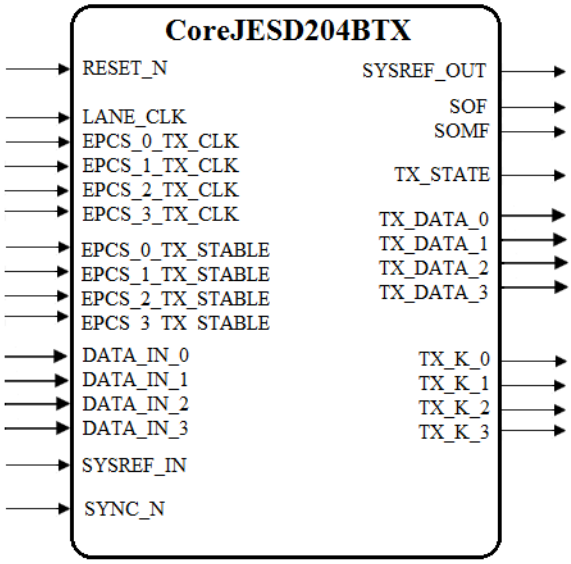


Table 8 • **CoreJESD204BTX I/O Signal Descriptions**

Name	Width	Direction	Description
RESET_N	1	In	This active low reset, resets the core to its initial state. This core has internal reset synchronizers per clock domain. For RTG4 family devices, RESET_N is used as a synchronous reset and to guarantee successful synchronization , this signal should be asserted for a minimum of 3 clock cycles of LANE_CLK frequency.
External PCS Transmitter Interface			
EPCS_[n]_TX_CLK	1	In	This is the clock for lane [n]. **Note:** [n] = 0 to 7
EPCS_[n]_TX_DATA	10, ((D_WIDTH/8)*10)	Out	EPCS transmitter channel [n] data. This output should match with the width of the EPCS receiver channel on the SERDES. If SERDES_MODE = 2 then Width = 10 If SERDES_MODE = 1 then Width = (D_WIDTH/8)*10 When ENCODER_EN=0 this signal is tied low. **Note:** [n] = 0 to 7
EPCS_[n]_TX_STABLE	1	In	EPCS transmitter channel [n] ready flag. This flag is 1 when the transmit PLL is locked to the reference clock. This is used along with RESET_N to reset the IP core. This IP core has internal Reset synchronizers per clock domain. **Note:** [n] = 0 to 7
8B10B Encoder Interface			
TX_DATA_[n]	8, D_WIDTH	Out	8b10b data for channel 0 data. This output should match with the width of the 8b10b encoder on the SERDES. If SERDES_MODE = 2 then Width = 8 If SERDES_MODE = 1 then Width = D_WIDTH When ENCODER_EN=1 this signal is tied low. **Note:** [n] = 0 to 7
TX_K_[n]	1, (D_WIDTH/8)	Out	8b10b K control for channel 0 data. If SERDES_MODE = 2 then Width = 1 If SERDES_MODE = 1 then Width = D_WIDTH/8 When ENCODER_EN=1 this signal is tied low. **Note:** [n] = 0 to 7
JESD204B Tx Interface			
DATA_IN_[n]	D_WIDTH	In	This is the input data to be transmitted on lane [n]. **Note:** [n] = 0 to 7
SYSREF_IN	1	In	This is the sysref input signal used by JESD204B subclass 1.

SYSREF_OUT	1	Out	This is the sysref output signal generated from inside the core. This or an external sysref signal is tied to SYSREF of the receiver or it can be used to loopback into the SYSREF_IN of the transmitter on a number of factors. (Subclass 1 only).
LANE_CLK	1	In	This is the clock all lanes will sync to internally in the IP core. Only when parameter/generic SERDES_MODE = 2 is when it differs from EPCS_*_TX_CLK. When SERDES_MODE = 2, EPCS_*_TX_CLK to LANE_CLK ratio's are as follows. 2:1 when D_WIDTH = 16bit 4:1 when D_WIDTH = 32bit 8:1 when D_WIDTH = 64bit When SERDES_MODE = 1, EPCS_*_TX_CLK to LANE_CLK ratio is 1:1 **Note:** It is recommended to generate this clock using Clock Conditioning Circuitry (CCC). Input clock to the CCC should be EPCS_0_TX_CLK.
SYNC_N	1	In	When asserted, the transmitter goes into its initial state and will start the ILA sequence. **Note:** The "_N" in the name represents active low and not the negative of a differential. When connecting to a differential input this signal must be connected to the positive rail on to the Y output of a INBUF_DIFF macro.
TX_STATE	2	Out	Monitors the current state of the transmitter. SYNC_ST INIT_LANE_ST DATA_ENC_ST Invalid state
SOF	(D_WIDTH/8)	Out	Monitors when start-of-frame occurs on all lanes.
SOMF	(D_WIDTH/8(	Out	Monitors when start-of-multi-frame occurs on all lanes.

Note: Different combinations of these IO's will be exposed depending on ENCODER_EN value.

5 Clocking and Reset

This section describes the options available for clocking and reset the IP core.

5.1 Clocking

Following clocks are used in the IP core.

EPCS_[n]_TX_CLK – EPCS transmitter channel [n] clock.

Note: [n] = 0 to 7

LANE_CLK - This is the clock to which all lanes are synchronized internally in the IP Core. This clock must be generated using EPCS_0_TX_CLK. For frequency relation of the LANE_CLK to EPCS_[n]_TX_CLK, please refer LANE_CLK description in Table 8.

Note: [n] = 0 to 7

5.2 Reset

Following resets are used in the IP core.

RESET_N - For RTG4 family, active low RESET_N is used as synchronous reset to reset the IP core. To guarantee successful synchronization, this signal should be asserted for a minimum of three clock cycles of LANE_CLK frequency.

For PolarFire/SmartFusion2/IGLOO2 families, active low RESET_N is used as asynchronous reset to reset the IP core.

EPCS_[n]_TX_STABLE - Similar to RESET_N, EPCS_[n]_TX_STABLE is used as synchronous reset to reset the IP core for RTG4 family and it is used as asynchronous reset for PolarFire/SmartFusion2/IGLOO2 families. This IP core has internal synchronizers per clock domain.

Note: [n] = 0 to 7

Following figure shows the reset synchronizer for RTG4 family.

Figure 5 • **Reset Synchronizer for EPCS_[n]_TX_CLK**

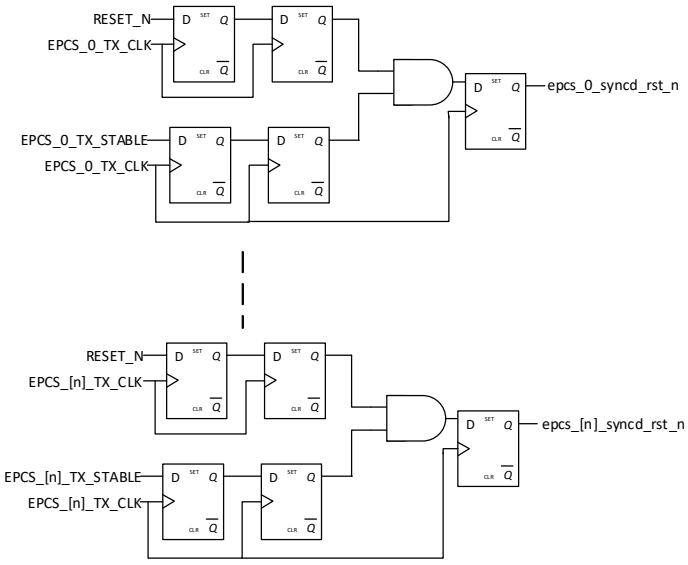
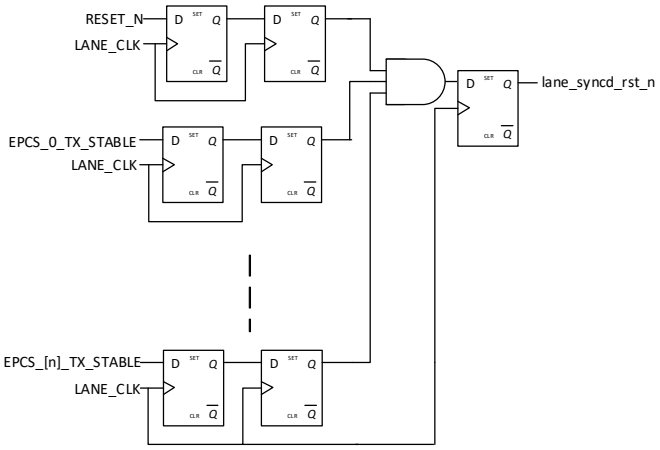


Figure 6 • **Reset synchronizer for LANE_CLK**



Following figure shows the reset synchronizer for PolarFire/SmartFusion2 and IGLOO2 families.

Figure 7 • **Reset Synchronizer for EPCS_[n]_TX_CLK**

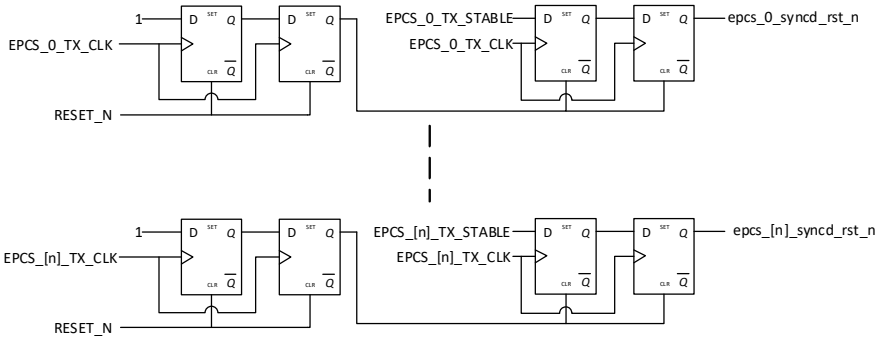
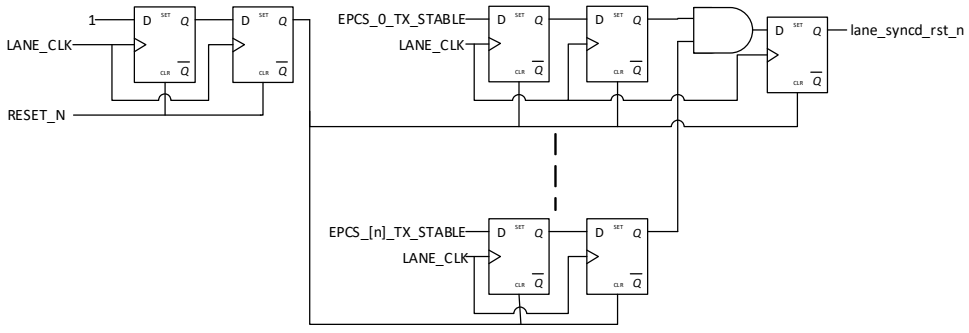


Figure 8 • **Reset Synchronizer for LANE_CLK**



Note: epcs_[n]_syncd_rst_n and lane_syncd_rst_n used as active low synchronous reset for RTG4 family and for PolarFire/SmartFusion2/IGLOO2 families they are used as active low asynchronous reset.

6 Tool Flow

6.1 License

This core will support generation of un-obfuscated Verilog and VHDL versions of the core. The un-obfuscated Verilog and VHDL versions will be license locked at the time of packaging. The core will be included in the Libero SoC IP bundle with Gold and Platinum licenses in clear RTL form.

6.2 SmartDesign

CoreJESD204BTX is pre-installed in the SmartDesign IP deployment design environment.

The core should be configured using the configuration GUI within the SmartDesign tool, as shown in Figure 11. For information on using SmartDesign to instantiate and generate cores, refer to the *Libero SoC Online Help* for instructions on obtaining IP documentation.

Figure 9 • **CoreJESD204BTX Full I/O View (4 lanes) with Encoder**

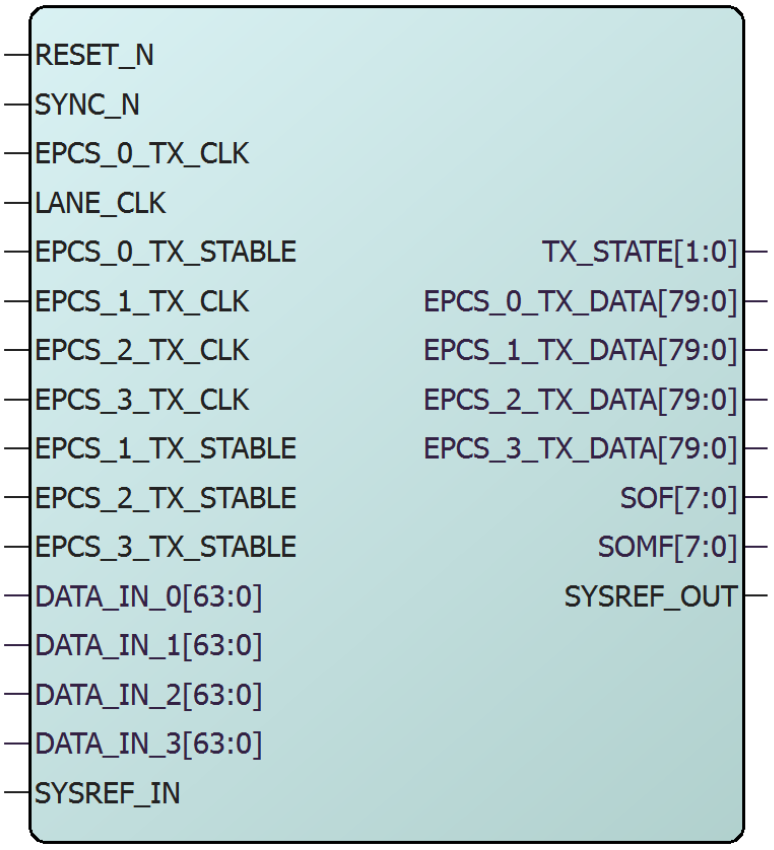
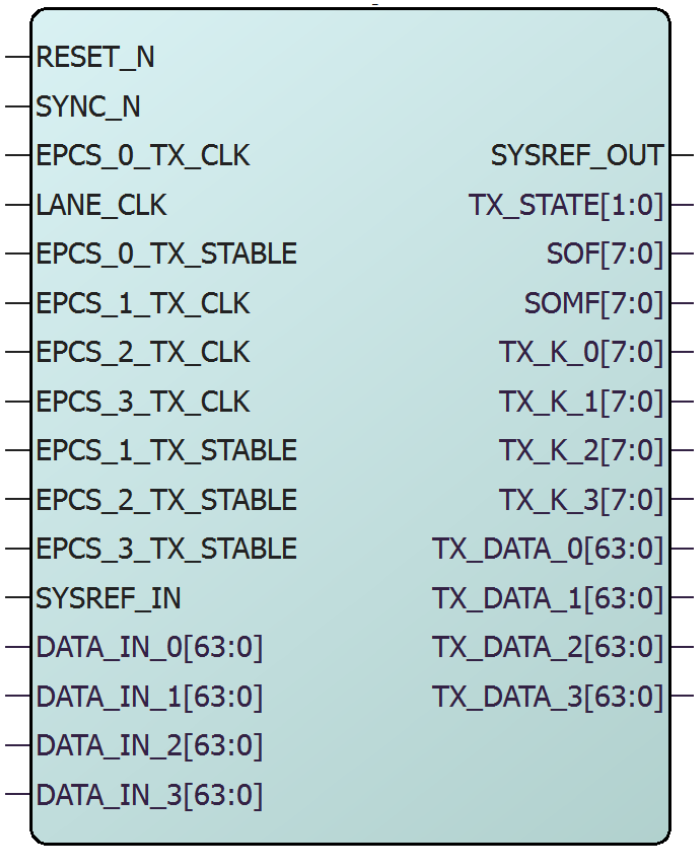


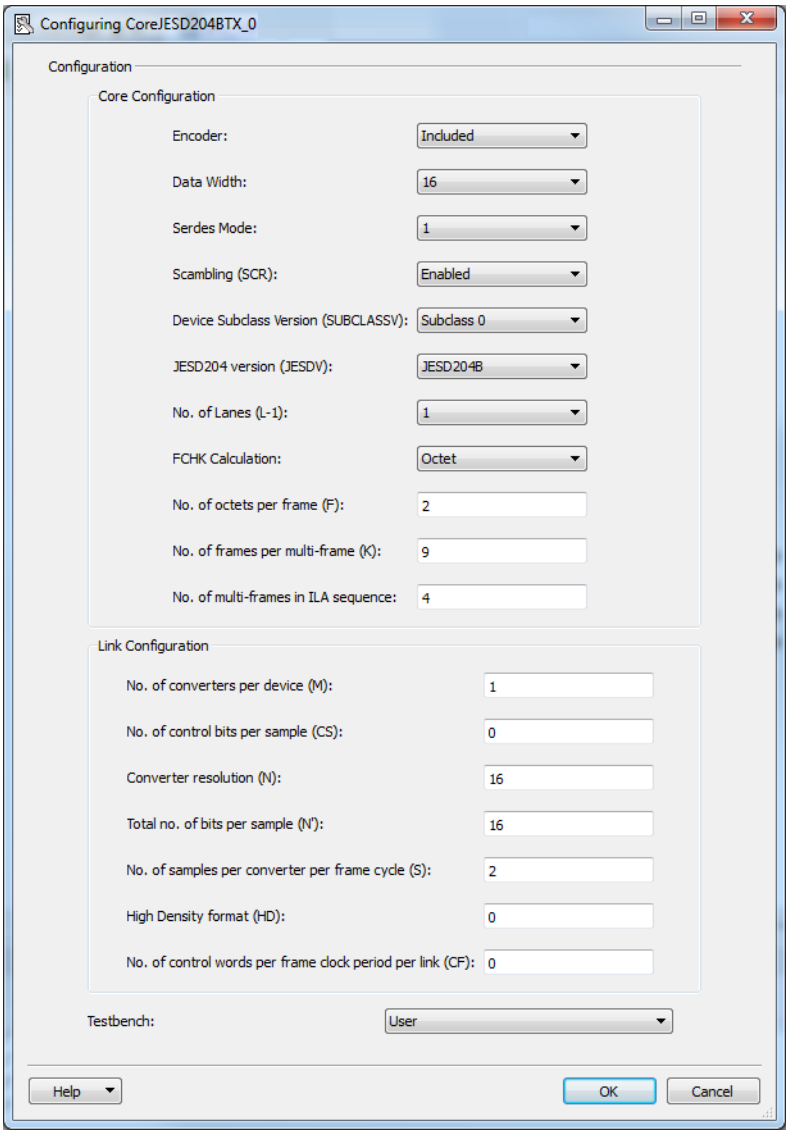
Figure 10 • **CoreJESD204BTX Full I/O View (4 lanes) without Encoder**



6.3 Configuring CoreJESD204BTX in SmartDesign

The core can be configured using the configuration GUI within SmartDesign. An example of the GUI for the SmartFusion2 family is shown in Figure 11.

Figure 11 • **CoreJESD204BTX SmartDesign Configuration GUI**



6.4 Simulation Flows

The User Testbench for CoreJESD204BTX is included in all releases.

To run simulations, select the **User Testbench** flow within the SmartDesign CoreJESD204BTX configuration GUI, right-click the canvas, and select **Generate Design**.

When SmartDesign generates the design files, it installs the user testbench files.

To run the user testbench, set the design root to the CoreJESD204BTX instantiation in the Libero SoC design hierarchy pane and click **Simulation** in the **Libero SoC Design Flow** window. This invokes ModelSim® and automatically runs the simulation.

6.5 Synthesis in Libero

After setting the design root appropriately for your design, click **Synthesis** in the Libero SoC software. The **Synthesis** window appears, displaying the Synplicity® project. Set Synplicity to VHDL 2008 standard if VHDL is being used. To run Synthesis, click **Run**.

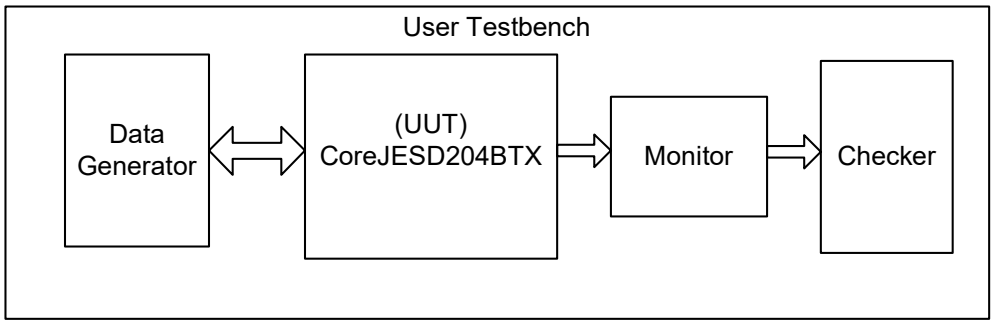
6.6 Place-and-Route in Libero

After setting the design root appropriately for your design, and running Synthesis, click **Layout** in the Libero SoC software to invoke Designer. CoreJESD204BTX requires no special place-and-route settings.

7 Testbench

CoreJESD204BTX user testbench gives an example of how to use the core.

Figure 12 • **CoreJESD204BTX User Testbench**



The simulation testbench shown in Figure 12 includes an instantiation of the CoreJESD204BTX macro, data generation, data monitor, and checker. Because the data on the output of the CoreJESD204BTX core is 8B10B encoded and scrambled, the user testbench is only used to test the core in a fixed configuration for each sub-class. The purpose of the testbench is to test the functionality of the core by inputting known data, monitoring the output, and checking for expected results.