

UG0701
User Guide
LiteFast IP



a  **MICROCHIP** company



a  MICROCHIP company

Microsemi 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

Email: sales.support@microsemi.com

www.microsemi.com

©2020 Microsemi, a wholly owned subsidiary of Microchip Technology Inc. All rights reserved. Microsemi and the Microsemi logo are registered trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), 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. Learn more at www.microsemi.com.

Contents

1	Revision History	1
1.1	Revision 5.0	1
1.2	Revision 4.0	1
1.3	Revision 3.0	1
1.4	Revision 2.0	1
1.5	Revision 1.0	1
2	LiteFast	2
2.1	Introduction	2
2.2	Key Features	3
2.3	Supported Device Families	3
2.4	References	3
3	Core Architecture	4
3.1	Design Description	4
3.1.1	Idle Frame	4
3.1.2	Data Frame	5
3.2	Flow Control	5
3.3	Multiple Lanes	6
4	Typical Application	8
4.1	Single Lane Application	8
4.2	Multiple Lanes Application	9
5	Interface Signals	11
5.1	Interface	11
5.2	Configuration Parameters	12
5.3	Key Interface Description	14
5.3.1	local_rece_rdy_tx_i	14
5.3.2	min_remote_token_tx_i	15
5.3.3	req_usr_data_tx_o and usr_data_tx_i	15
5.4	Timing Diagram	15
5.5	Resource Utilizations	16

Figures

Figure 1	Typical Application for LiteFast	2
Figure 2	Idle Frame Structure	4
Figure 3	Data Frame Structure	5
Figure 4	Token Byte Transfer	6
Figure 5	Striping Process for Four Lane Application	7
Figure 6	Application for Single Lane	8
Figure 7	Application for Two Lanes	9
Figure 8	Application for Four Lanes	10
Figure 9	LiteFast Interface Signal	11
Figure 10	Timing Diagram for LiteFast Transmitter	15

Tables

Table 1	Configuration Parameters	12
Table 2	User Application Data Width	12
Table 3	LiteFast Transmitter Interface Signals	13
Table 4	LiteFast Receiver Interface Signals	14
Table 5	min_remote_token_tx_i settings	15
Table 6	Resource Utilization for RTG4 FPGAs	16
Table 7	Resource Utilization for SmartFusion2 SoC FPGAs	16
Table 8	Resource Utilization for PolarFire FPGAs	16

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 5.0

The following is a summary of the changes in revision 5.0 of this document.

- Supports 32 bits on a single lane.
- Updated [Key Features](#), page 3.
- Updated [Table 1](#), page 12, [Table 2](#), page 12, [Table 5](#), page 15, and [Table 8](#), page 16.

1.2 Revision 4.0

In revision 4.0 of this document, resource utilization information for PolarFire FPGAs was updated. For more information, see [Table 8](#), page 16.

1.3 Revision 3.0

The following is a summary of changes made in revision 3.0 of this document

- PolarFire references are added where ever necessary in the document. PolarFire is listed in the FPGA families supported by the IP core in [Supported Device Families](#), page 3.
- Two new documents, UG0677: PolarFire FPGA Transceiver User Guide and DG0783: PolarFire FPGA: High-Speed Data Transfer In 8b10B Mode Using the LiteFast IP Demo Guide are added with their hyperlinks in [References](#), page 3.
- Content is edited and few existing details are removed in [Introduction](#), page 2, [Single Lane Application](#), page 8 and [Multiple Lanes Application](#), page 9 sections.
- A new table, [Table 8](#), page 16 is added to list the resource utilization of the LiteFast IP implemented in the PolarFire devices.

1.4 Revision 2.0

In revision 2.0 of this document, resource utilization information for RTG4 FPGAs was added. For more information, see [Table 6](#), page 16.

1.5 Revision 1.0

Revision 1.0 was the first publication of this document.

2 LiteFast

2.1 Introduction

There are many serial protocols for high-speed, data-intensive applications. In order to provide low-cost, scalable, light weight, and high-speed solutions, Microsemi has designed the LiteFast serial protocol specifically for the next generation of Microsemi FPGA devices.

Similar to other serial protocols, LiteFast has in-built flow control and other features to maintain link activity when the applications are not involved in data transfer.

The LiteFast transmitter embeds application data within data frames and initiates data transfer. The LiteFast receiver extracts the application data from the data frame and delivers it to the user interface. When there is no application data for transmission, an idle frame is transmitted to maintain the physical link between systems.

For a targeted application of a given system, the data extracted from a data frame is written into a receiver buffer. If the available storage space in the receiver buffer approaches zero, LiteFast notifies the remote LiteFast transmitter to pause data frame transmission and to prevent buffer overflow. When the available storage space returns to a value greater than a given threshold value, the LiteFast receiver notifies the remote LiteFast transmitter to resume data frame transmission. The threshold value should be at least 128 bytes, and the upper limit of the threshold is fixed by the user application.

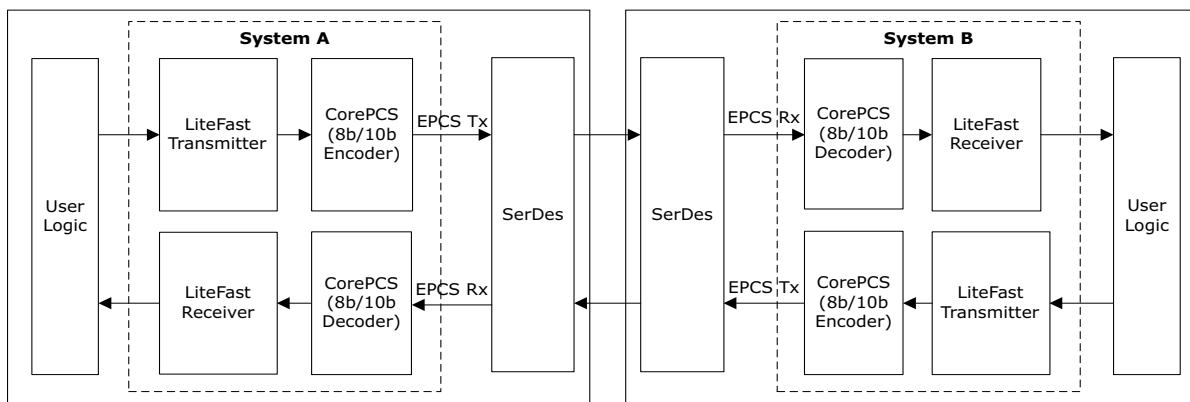
LiteFast supports x1, x2, or x4 lanes per SerDes. When multiple lanes are used, the data transferring bandwidth between two systems can be greater than the bandwidth of a single SerDes lane maximal data transferring bandwidth.

In the following figure, application data flows from System A to System B through a pair of high-speed SerDes lanes, to show how the system works.

In System A, LiteFast transmitter embeds the application data from user logic into the data frame, it also generates an idle frame when there is no application data. Data frame and idle frame are encoded in 8b10b encoder module. The data is sent to System B through the high-speed SerDes lane.

In System B, the parallel data stream from the SerDes back end receiver is decoded in the 8b10b decoder module. The LiteFast receiver recognizes the data frame and idle frame, extracts application data from the data frame payload, and drops the idle frame.

Figure 1 • Typical Application for LiteFast



2.2 Key Features

The following are the main features of the LiteFast IP:

- Idle frame to establish and maintain the link and data frame for user data
- Flow control through token exchange
- Supports x1, x2, or x4 per SerDes
- Supports cumulative speeds from 4 to 10 Gbps for x4 lanes, 20 Gbps for x2 lanes (10 Gbps per lane), and 12.7 Gbps for x1 lane.
- Word alignment, block alignment, and lane alignment for the receive chain
- Independent of user application and the device
- Serial full duplex or serial simplex operation
- Supports CRC-32
- Supports hot plug
- Data packet size: 1 to 128 bytes of application data. The length of the payload must be a multiple of eight, otherwise K28.4 bytes are filled to meet the requirement
- Idle packet: 8 bytes
- Supports 8b10b encoding mechanism
- Supports for little endian

2.3 Supported Device Families

The following FPGA families are supported by the IP core:

- SmartFusion[®]2 (All devices that have transceiver)
- IGLOO[®]2 (All devices that have transceiver)
- RTG4[™]
- PolarFire[®]

Note: Ensure that the selected device in the family has transceivers.

2.4 References

The following documents are referred in this user guide.

- *SmartFusion2 and IGLOO2 High Speed Serial Interface Configuration*
- *CoreUART Handbook*
- *CorePCS Handbook*
- *UG0541: SmartFusion2 SoC FPGA Evaluation Kit User Guide*
- *DG0720: LiteFast IP Demo Guide*
- *DG0729: LiteFast IP on RTG4 Demo Guide*
- *UG0567: RTG4 FPGA High Speed Serial Interfaces User Guide*
- *UG0677: PolarFire FPGA Transceiver User Guide*
- *DG0783: PolarFire FPGA: High-Speed Data Transfer In 8b10B Mode Using the LiteFast IP Demo Guide*

3 Core Architecture

3.1 Design Description

LiteFast uses idle frame for link establishment and data frame for data transfer.

3.1.1 Idle Frame

To enable point-to-point communication, the LiteFast transmitter uses Idle frames to establish a link. After the serial link is up, LiteFast transmitter embeds application data into data frames and sends it to the remote LiteFast receiver. When there is no data transmission, LiteFast IP uses Idle frame to maintain the link.

The following figure shows the Idle frame structure.

Figure 2 • Idle Frame Structure

/I/	/I/	/I/	/I/	/I/	/I/	/I/	Token
-----	-----	-----	-----	-----	-----	-----	-------

Token is one byte. // character can be one of the following characters:

- /K/ character (K28.5)
- /R/ character (K28.0)
- /A/ character (K28.3)

Following are the rules for // character:

- When multiple lanes are transmitting // characters simultaneously, // on all lanes must be the same character.
- /A/ character must be the first byte of the idle frame.
- There must be at least 31 bytes between two /A/ characters.
- If a // character is not /A/ character, it must be /K/ or /R/ character randomly.

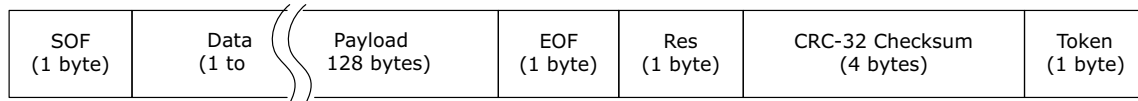
When multiple lanes are used for data transfer, the LiteFast receiver aligns all the lanes by using the /A/ character. The receiver may also use /K/ bytes to align the 10 bits word.

Flow control is established by token exchange.

3.1.2 Data Frame

Application data is transmitted using data frames. The following figure shows the data frame structure.

Figure 3 • Data Frame Structure



Data frame fields include the following fields:

- Start of frame (SOF): Includes one K28.1 byte. The LiteFast receiver recognizes the header of the data frame by its SOF field.
- Data payload: Contains 1 to 128 bytes of application data. The length of the payload must be the multiple of 8, otherwise PAD (K28.4) bytes are filled to meet this requirement.
- End of frame (EOF): Includes one K28.7. LiteFast receiver recognizes the end of the data payload by EOF field.
- Reserved (Res): Reserved segment is 1-byte.
- CRC-32 check sum: Contains the CRC-32 check sum for data payload. The polynomial is as shown below

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

EQ1

- Token: It is one byte, which indicates the available buffer in the receive chain.

3.2 Flow Control

In order to avoid a buffer overflow in receive chain, the local receiver communicates the available buffers to the local transmitter. Local transmitter communicates the same to the remote receiver through tokens. The remote LiteFast transmitter pauses data frame transmission when local receiver buffer is almost full, and the remote LiteFast transmitter resumes data frame transmission when the receiver buffer is not almost full.

The receiver buffer's available storage space is encoded to token byte, which ranges from 0 to 255 bytes. Token is set to 1 when there are 0 to 127 available bytes in the receiver buffer. The token is set to two when there are 128 to 255 available bytes in the receiver buffer. The rule of token calculation is shown in the following equation.

$$\text{Token} = \text{Int} \frac{\text{Rx Buffer available bytes}}{128}$$

EQ2

The token field indicates the number of bytes available in the remote receiver. The remote transmitter pauses data frame transmission, if the local receiver buffer is not able to store more than one data frame.

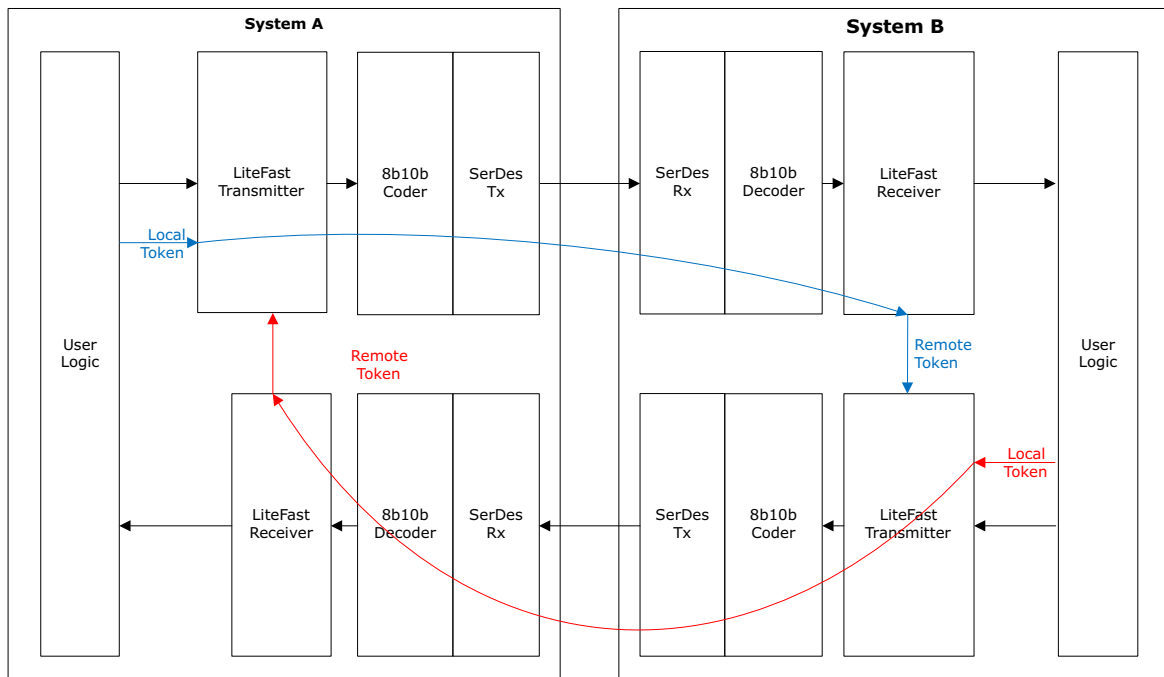
In Figure 4, page 6 System A and System B utilize SerDes and LiteFast to transfer data over a serial link.

In System A, the available storage space of the receiver buffer is encoded to a token byte. Token byte is provided to the LiteFast transmitter module and padded at the end of the data frame and idle frame.

In System B, LiteFast receiver receives the remote token from System A. The LiteFast transmitter determines whether it should send a LiteFast data frame according to the value of the remote token byte. System B also sends the local token byte to System A, the flow control mechanism is implemented by transferring token bytes on both sides.

The following figure shows the flow control mechanism.

Figure 4 • Token Byte Transfer

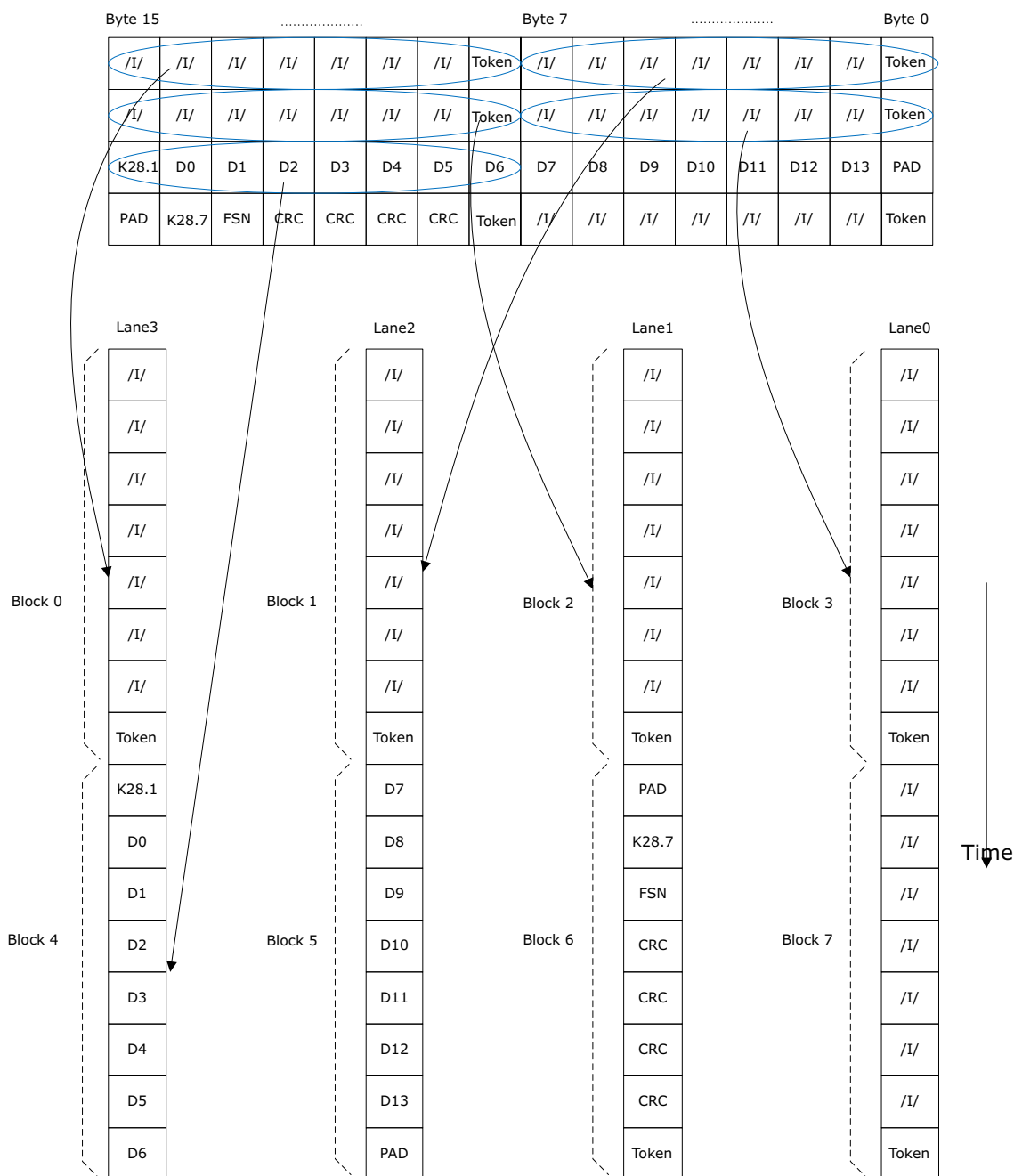


3.3 Multiple Lanes

In multiple lanes, application data frame blocks are transmitted and received over multiple lanes:

- Data stream is cut into data blocks, each data block includes 8 bytes
- Idle frame is one data block

The following figure shows the stripping process for a four lane application. The data block is distributed on different lanes as blocks 0, 1, 2, 3. In this example, block 0 is placed on lane 3, block 1 on lane 2, block 2 on lane 1, and block 3 on lane 0.

Figure 5 • Stripping Process for Four Lane Application

In the LiteFast receiver, all lanes are aligned using /A/ characters, and the LiteFast data stream is recovered using a reversed stripping operation.

After the system is reset, LiteFast sends LiteFast idle frames to establish a link between the systems. The LiteFast receiver performs lane alignment using the Idle frames.

The data direction is from Fabric to SerDes.

4 Typical Application

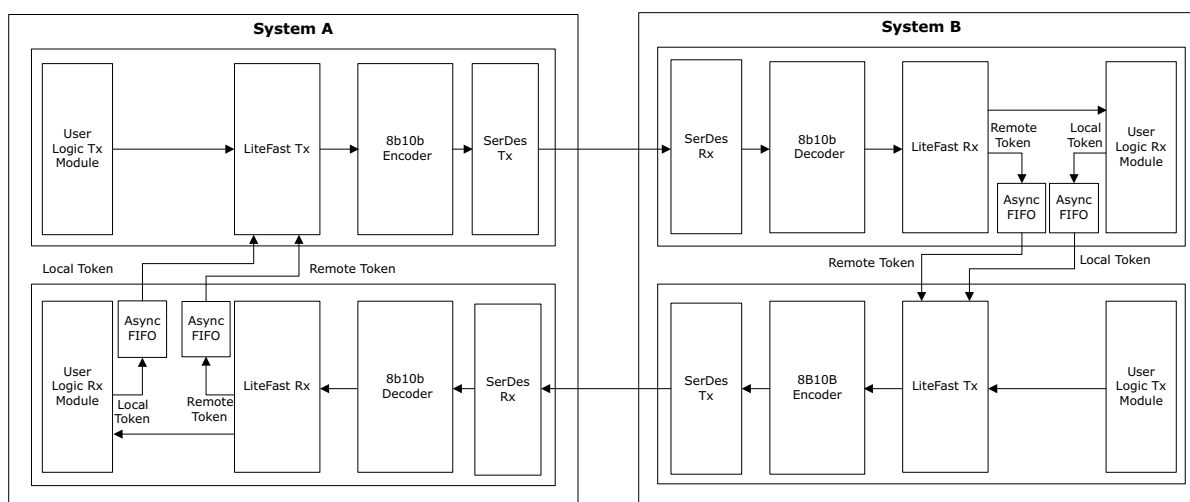
4.1 Single Lane Application

In the following figure, System A and System B transmit data through a pair of SerDes.

In System A, the LiteFast transmitter embeds user data into LiteFast data frames, and it also generates a LiteFast idle frame to build and maintain the link between two systems. The LiteFast data stream is encoded in 8b10b encoder and is sent to System B through the SerDes transmitter.

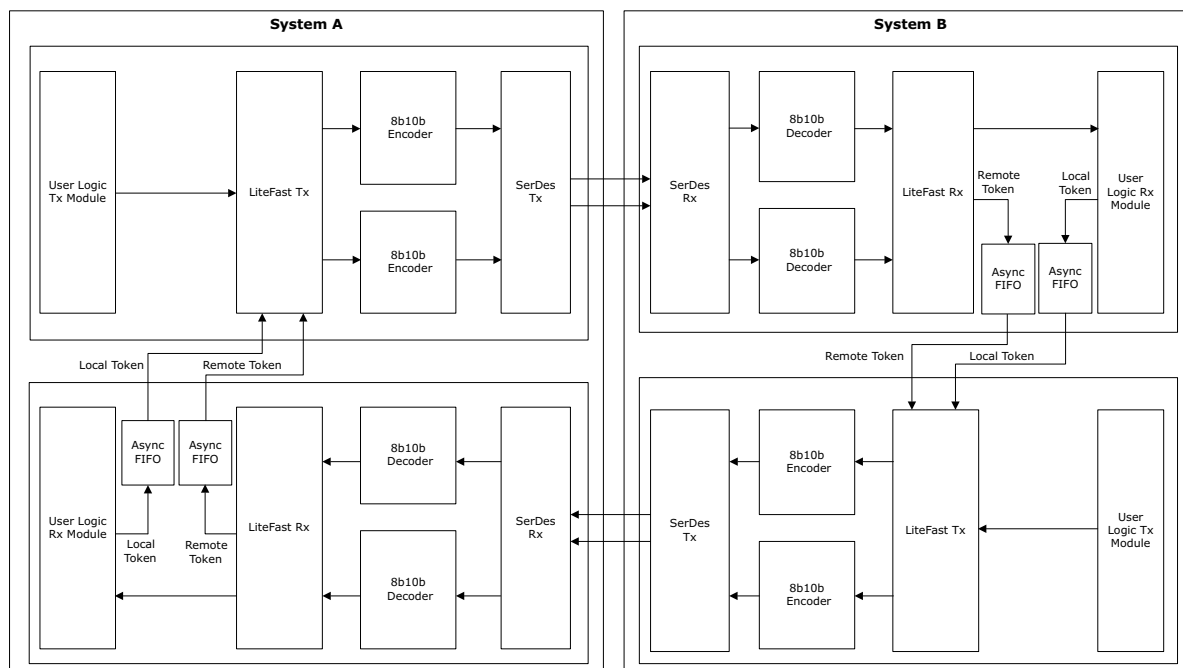
In the System A receiver, parallel data from SerDes back end is decoded in 8b10b decoder. LiteFast receiver recognizes the idle frame and data frame and writes the data frame payload into RX FIFO (first-in-first-out). Remote Token information is extracted from the LiteFast frames. From the perspective of System A, Remote Token information indicates the available storage space in System B Rx FIFO, and the local token indicates the available storage space in System A RX FIFO. The following figure shows, LiteFast transmitter and LiteFast receiver work in different clock domains.

Figure 6 • Application for Single Lane



4.2 Multiple Lanes Application

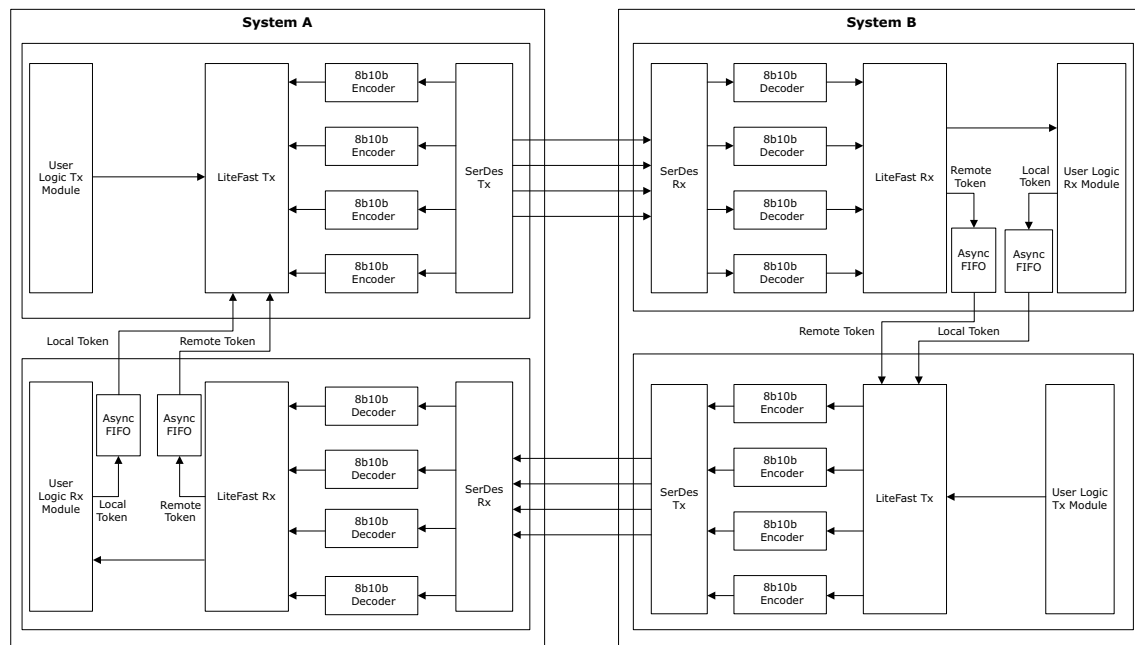
Figure 7 • Application for Two Lanes



In the following figure, System A and System B transmit data through multiple pairs of SerDes.

In System A, LiteFast transmitter embeds user data into LiteFast data frames, and generates LiteFast idle frame to establish and maintain the link between two systems. LiteFast data stream is stripped on multiple lanes, encoded in 8b10b encoder, and sent to System B through multiple SerDes transmitters. Each SerDes transmitter works in its own clock domain, a group of asynchronous FIFO are set for data synchronization between 8b10b encoder and SerDes. All clocks in System A transmitter must be from the same clock source.

In System A receiver, parallel data from SerDes back end is decoded in 8b10b decoder. LiteFast performs lane alignment and recovers LiteFast data stream by using reversed stripping process. LiteFast receiver recognizes the idle frame, data frame, and write data frame payload into RX FIFO, Remote Token information is extracted from the LiteFast frames. From the perspective of System A, Remote Token information indicates the available storage space in System B Rx FIFO, and Local Token indicates the available storage space in the System A Rx FIFO. LiteFast transmitter and LiteFast receiver work in different clock domain, Local Token and Remote Token are synchronized through asynchronous FIFO. Each SerDes receiver works in its own clock domain, a group of asynchronous FIFO are set for data synchronization between 8B10B decoder and SerDes receiver. All clocks in System A receiver should be from the same clock source.

Figure 8 • Application for Four Lanes

Users can implement application specific user logic interfaced with the LiteFast.

Note: The LiteFast does not include 8b/10b encoder and decoder blocks. These blocks can be instantiated from the Libero[®] catalog. LiteFast is device independent.

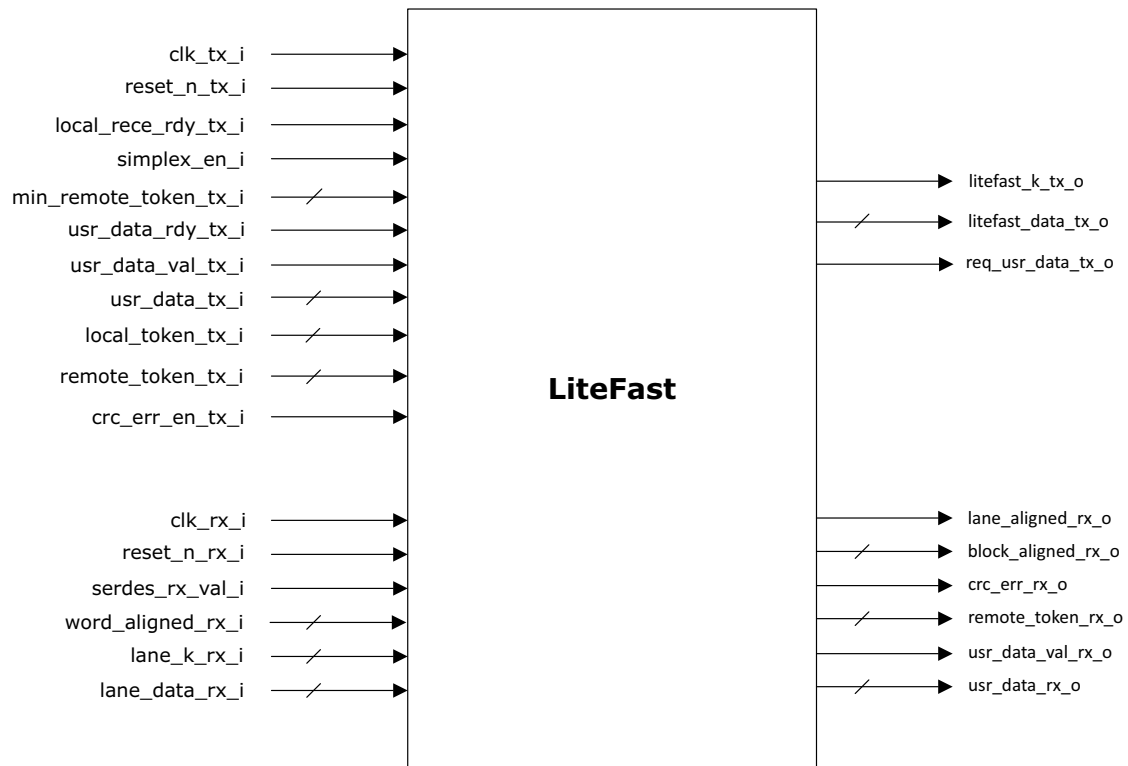
Local and Remote Token exchanges happen across multiple clock domains. Therefore, care should be taken for clock domain crossing.

5 Interface Signals

5.1 Interface

LiteFast includes the LiteFast transmitter and LiteFast receiver. The following figure shows the block diagram of LiteFast interface signals.

Figure 9 • LiteFast Interface Signal



5.2 Configuration Parameters

The following table lists the configuration parameters used in the hardware implementation process of LiteFast.

Table 1 • Configuration Parameters¹

Name	Description
g_DATA_WID	LiteFast data bus width, it can be 8 bits, 16 bits, 32 bits, or 64 bits.
g_LANE_NUM	Lane number. It can be 1, 2, or 4. When g_DATA_WID = 8, it must be 1. When g_DATA_WID = 16, it may be 1 or 2. When g_DATA_WID = 32, it may be 1, 2, or 4. When g_DATA_WID = 64, it may be 2 or 4.

1. 8-bit width is not supported with 8b10b mode in XCVR on PolarFire devices.

Note: Parameters are generic and vary based on the application requirements.

The following table lists the user application data width for LiteFast in SmartFusion2, IGLOO2, RTG4, and PolarFire devices.

Table 2 • User Application Data Width¹

User Application Data Width	1 Lane	2 Lanes	4 Lanes
8 bits	Support	No	No
16 bits	Support maximum 16 bits in one lane	Support maximum 8 bits per lane	No
32 bits	Support maximum 32 bits per lane	Support maximum 16 bits per lane	Support maximum 8 bits per lane
64 bits	No	Support maximum 32 bits per lane	Support maximum 16 bits per lane

1. 8-bit width is not supported with 8b10b mode in XCVR on PolarFire devices.

The following table lists the description of input and output ports for LiteFast transmitter interface signals.

Table 3 • LiteFast Transmitter Interface Signals

Interface	Input /Output	Width	Description
reset_n_tx_i	Input	1	Low active reset signal.
clk_tx_i	Input	1	Clock.
local_rece_rdy_tx_i	Input	1	Indicates that the local LiteFast receiver is ready to receive the data frame, it means SerDes/8b10b decoder are in normal working mode and multiple lanes are aligned.
simplex_en_i	Input	1	Disables flow control feature.
min_remote_token_tx_i	Input	[7:0]	LiteFast transmitter is allowed to send data frame only when remote token is greater or equal to this signal value.
usr_data_rdy_tx_i	Input	1	Indicates that the user logic module has data to be transmitted through LiteFast.
usr_data_val_tx_i	Input	1	Indicates whether data on usr_data_tx_i port is valid.
usr_data_tx_i	Input	[g_DATA_WID -1:0]	Data from user module, data on this port is wrapped into the LiteFast data frame.
local_token_num_tx_i	Input	[7:0]	Indicates the available storage space in local receiver buffer.
remote_token_num_tx_i	Input	[7:0]	Indicates the available storage space in remote receiver buffer.
crc_err_en_tx_i	Input	1	Enables CRC error insertion, only for debug purpose.
req_usr_data_tx_o	Output	1	Requires user data from user module.
litefast_k_tx_o	Output	[g_DATA_WID / 8) - 1:0]	Indicates whether the byte on LiteFast_data_tx_o port is an 8b10b special character. Bit[0] indicates the meaning for LiteFast_data_tx_o[7:0].
litefast_data_tx_o	Input	[g_DATA_WID -1:0]	LiteFast data stream. In multiple lanes mode, if lane data width is 8 bits, bit[7:0] belongs to Lane 0 and if lane data width is 16 bits, bit[15:0] belongs to Lane 0.

Note: Two clock cycle delay is required between req_usr_data_tx_o and usr_data_val_tx_i, as shown in [Figure 10](#), page 15.

The following table lists the description of input and output ports for LiteFast receiver interface signals.

Table 4 • LiteFast Receiver Interface Signals

Interface	Input/ Output	Width	Description
reset_n_rx_i	Input	1	Low active reset signal.
clk_rx_i	Input	1	Clock.
serdes_rx_val_i	Input	[g_LANE_NUM -1:0]	Indicates whether SerDes receiver is ready for work.
word_aligned_rx_i	Input	[g_LANE_NUM -1:0]	Indicates whether 8b10b decoder is ready for work. In multiple lanes mode, bit[0] indicates the status for Lane 0.
lane_k_rx_i	Input	[g_DATA_WID / 8) - 1:0]	Indicates whether byte on lane_data_rx_i port is an 8b10b special character. Bit[0] indicates the meaning for lane_data_rx_i[7:0].
lane_data_rx_i	Input	[g_DATA_WID - 1:0]	Input data stream. In multiple lanes mode, if lane data width is 8 bits, bit[7:0] belongs to Lane 0 and if lane data width is 16 bits, bit[15:0] belongs to Lane 0.
crc_err_rx_o	Output	1	Indicates whether CRC error is received.
usr_data_val_rx_o	Output	1	Indicates whether data on usr_data_rx_o port is valid.
usr_data_rx_o	Output	[g_DATA_WID -1:0]	User data from the LiteFast data frame payload.
remote_token_rx_o	Output	[7:0]	Remote token byte.
lane_aligned_rx_o	Output	1	Indicates whether multiple lanes are aligned and whether SerDes and 8b10b decoder are working fine.
block_aligned_rx_o	Output	[g_LANE_NUM -1:0]	Indicates whether each lane's data block is aligned. It is only for debug purpose. Bit[0] indicates the status of Lane 0.

5.3 Key Interface Description

5.3.1 local_rece_rdy_tx_i

The interface port local_rece_rdy_tx_i in the LiteFast transmitter must be connected with the local LiteFast receiver's lane_aligned_rx_o port.

5.3.2 min_remote_token_tx_i

For the purpose of flow control, LiteFast transmitter pauses data frame transmission when remote token is less than min_remote_token_tx_i. When setting the value of min_remote_token_tx_i, the maximal remote token transmission delay and maximal delay must be considered between the LiteFast transmitter and remote receiver buffer.

If user uses the integration design in typical application chapter, the transmission delay between SerDes receiver pin and SerDes transmitter pin is less than 10 ns. The following table lists the min_remote_token_tx_i setting details.

Table 5 • min_remote_token_tx_i settings

Data Width	Lane Number	min_remote_token_tx_i
g_DATA_WID = 8	g_LANE_NUM = 1	3
g_DATA_WID = 16	g_LANE_NUM = 1	6
	g_LANE_NUM = 2	6
g_DATA_WID = 32	g_LANE_NUM = 1	8
	g_LANE_NUM = 2	8
	g_LANE_NUM = 4	8
g_DATA_WID = 64	g_LANE_NUM = 2	16
	g_LANE_NUM = 4	16

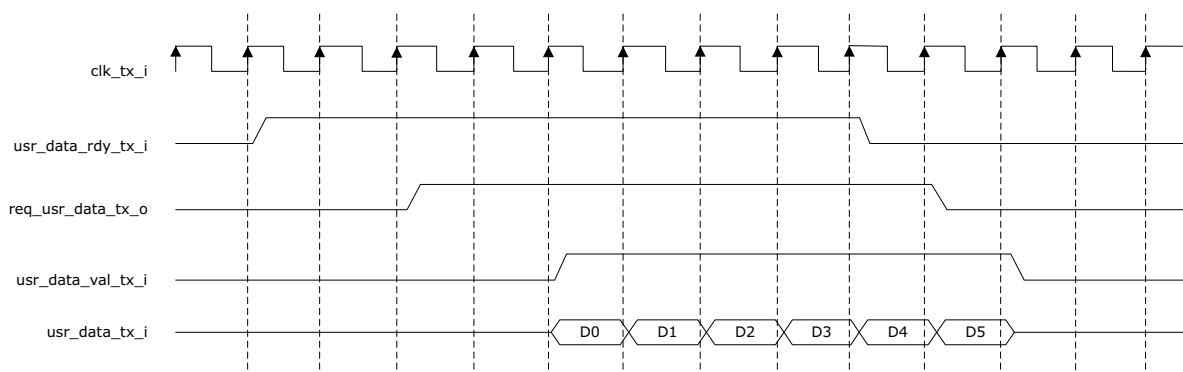
5.3.3 req_usr_data_tx_o and usr_data_tx_i

LiteFast Transmitter requires data from the user logic module.

5.4 Timing Diagram

The following figure shows the timing operation of the LiteFast transmitter.

Figure 10 • Timing Diagram for LiteFast Transmitter



5.5 Resource Utilizations

The following table lists the resource utilization of the LiteFast IP implemented in a RTG4 FPGA device.

Table 6 • Resource Utilization for RTG4 FPGAs

Working Mode	4LUT	DFF	RAM64x18	RAM1K18
8 bits x 1 lane	338	298	0	0
8 bits x 2 lane	1334	1168	2	0
8 bits x 4 lane	2377	2012	4	0
16 bits x 1 lane	531	393	0	0
16 bits x 2 lane	1675	1451	4	0
16 bits x 4 lane	2190	2535	8	0

The following table lists the resource utilization of the LiteFast IP implemented in the SmartFusion2 SoC FPGA device.

Table 7 • Resource Utilization for SmartFusion2 SoC FPGAs

Working Mode	SLE	CFG	RAM64x18	RAM1K18
8 bits x 1 Lane	300	310	0	0
8 bits x 2 Lanes	1060	1250	2	0
8 bits x 4 Lanes	1850	2300	4	0
16 bits x 1 Lane	380	480	0	0
16 bits x 2 Lanes	1260	1450	4	0
16 bits x 4 Lanes	2220	2540	8	0

The following table lists the resource utilization of the LiteFast IP implemented in the PolarFire devices.

Table 8 • Resource Utilization for PolarFire FPGAs

Working Mode	SLE	CFG	μSRAM	LSRAM
16 bits x 1 Lane	393	515	0	0
16 bits x 2 Lanes	1313	1576	4	0
16 bits x 4 Lanes	2230	2464	8	0
32 bits x 1 Lane	502	820	0	0
32 bits x 2 Lanes	1659	1889	8	0

Note: The preceding utilization values must be used for approximate estimates only. Utilization varies in actual implementation.