



# AN4792

---

---

## PolarFire® FPGA High-Speed Data Transfer in 8b10b Mode Using the LiteFast IP

---

---

### Introduction

---

This document describes how to run the LiteFast IP demo on the PolarFire® Evaluation Board using the LiteFast GUI application. The GUI application is packaged along with the design files. The reference design is built using the PolarFire high-speed transceiver block in 8b10b mode and the LiteFast IP core. It operates in loopback mode because the TX and RX transceiver lanes are manually looped back on the board. This setup facilitates a standalone demo that does not require another board.

## Table of Contents

Introduction.....	1
1. High-Speed Data Transfer Using the LiteFast IP.....	3
1.1. Design Requirements.....	3
1.2. Prerequisites.....	4
1.3. Demo Design.....	4
1.4. Clocking Structure.....	17
1.5. Simulating the Design.....	17
2. Libero Design Flow.....	22
2.1. Synthesize.....	22
2.2. Place and Route.....	22
2.3. Verify Timing.....	23
2.4. Generate FPGA Array Data.....	24
2.5. Design and Memory Initialization.....	24
2.6. Generate Bitstream.....	24
2.7. Run PROGRAM Action.....	24
3. Running the Demo.....	26
4. Using LiteFast For Board-to-Board Data Transfer.....	30
5. Appendix 1: Programming the Device Using FlashPro Express.....	31
6. Appendix 2: Running the TCL Script.....	33
7. Appendix 3: References.....	34
8. Revision History.....	35
Microchip FPGA Support.....	36
Microchip Information.....	36
The Microchip Website.....	36
Product Change Notification Service.....	36
Customer Support.....	36
Microchip Devices Code Protection Feature.....	36
Legal Notice.....	37
Trademarks.....	37
Quality Management System.....	38
Worldwide Sales and Service.....	39

## 1. High-Speed Data Transfer Using the LiteFast IP

Microchip's LiteFast IP core implements a serial, point-to-point, and light-weight protocol for high-speed serial communication. LiteFast IP creates a high-speed serial link by connecting to the transceiver block available in Microchip's PolarFire device. The high-speed transceiver block handles data rates ranging from 250 Mbps to 12.7 Gbps. The transceiver (PF\_XCVR) module integrates several functional blocks to support high-speed serial data transfer within the FPGA.

The LiteFast IP supports data widths of 8, 16, 32, and 64 bits and supports multiple transceiver lanes. In the reference design, the LiteFast IP is configured to 32-bit data width and single lane.

For more information about the LiteFast design implementation, and the necessary blocks and IP cores instantiated in Libero® SoC PolarFire, see [1.3. Demo Design](#).

The reference design can be programmed using any of the following options:

- **Using the pre-generated .job file:** To program the device using the .job file provided along with the reference design, see [5. Appendix 1: Programming the Device Using FlashPro Express](#).
- **Using Libero SoC PolarFire:** To program the device using Libero SoC PolarFire, see [2. Libero Design Flow](#).

The reference design can be used on two Microchip PolarFire boards to implement a full-duplex data transfer application. For more information about the implementation of LiteFast IP for data transfer between two boards, see [4. Using LiteFast For Board-to-Board Data Transfer](#).

### 1.1 Design Requirements

The following table lists the hardware and software design requirements for running this demo design.

**Table 1-1. Design Requirements**

Requirement	Version
Operating System	Windows® 10
<b>Hardware</b>	
PolarFire Evaluation Kit (MPF300-EVAL-KIT) – PolarFire Evaluation board – 12V/5A wall-mounted power adapter – USB 2.0 A-male to mini-B cable for UART and programming	Rev B or later
2 SMA-to-SMA cables (not provided with the kit)	—
<b>Software</b>	
FlashPro® Express	Refer to the <code>readme.txt</code> file provided in the design files for the software versions used with this reference design.
Libero® SoC PolarFire	
ModelSim®	
Synplify Pro®	
<b>IP</b>	
LiteFast IP core	—
PF XCVR IP core	—
PF_TX_PLL	—
PF_XCVR_REF_CLK	—

.....continued	
Requirement	Version
CoreUART	—
COREFIFO	—



**Important:** Libero SmartDesign and configuration screen shots shown in this guide are for illustration purpose only. Open the Libero design to see the latest updates.

## 1.2 Prerequisites

Before you start:

1. Download the reference design files from the following location: [www.microchip.com/en-us/application-notes/AN4792](http://www.microchip.com/en-us/application-notes/AN4792).
2. Download and install Libero SoC PolarFire v2022.2 on the host PC from the following location: [Libero SoC Documentation](#).  
The latest versions of ModelSim and Synplify Pro are included in the Libero SoC PolarFire installation package.

## 1.3 Demo Design

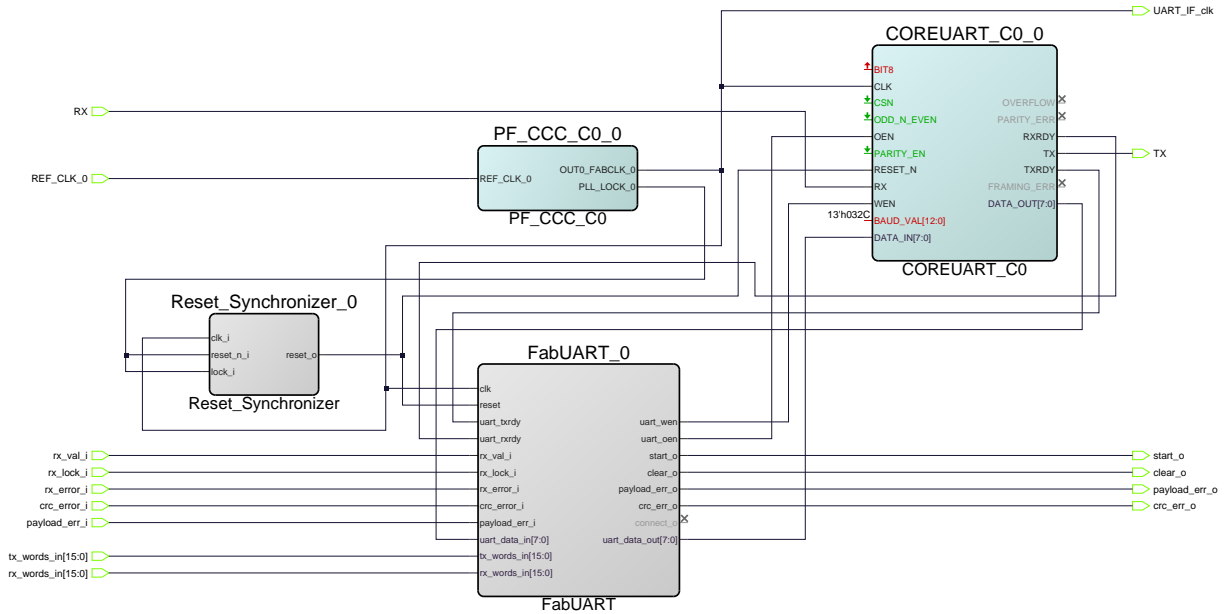
In the reference design:

1. The UART\_IF\_0 block interfaces with the GUI. This block receives RX signals to start and stop the LiteFast demo. This block drives the Counter\_0 and the Count\_Checker\_0 blocks when the start signal is received. When a CRC error or payload error is selected on the GUI, the UART\_IF\_0 block receives that RX signal and passes it to the Counter\_0 block for error injection.
2. The Counter\_0 block acts as the application that transfers 32-bit parallel data to the LiteFast\_tx\_0 block.
3. The LiteFast\_tx\_0 block is the instantiation of LiteFast IP, configured as transmitter. It receives the 32-bit data, converts the data to LiteFast frames and forwards the data to the PF\_XCVR\_0 block.
4. The PF\_XCVR\_0 (transceiver) IP block receives the data on its TX lane, encodes the data in 8b10b format, and serializes the data. The encoded and serialized data is looped back to the RX lane.
5. PF\_XCVR\_0 decodes the data in 8b10b format, deserializes the data on its RX lane, and then sends the decoded data to the LiteFast\_rx\_0 block, which is the instantiation of LiteFast IP configured as receiver.
6. The Count\_Checker\_0 block generates 32-bit data in sync with Counter\_0, and compares this data with the 32-bit data received from the LiteFast\_rx\_0 block. This block also sends the number of TX words transmitted, status of serial link, CRC error, and payload error to the UART\_IF\_0 block.
7. The UART\_IF\_0 block forwards these status and error information on its TX interface to the GUI for display.



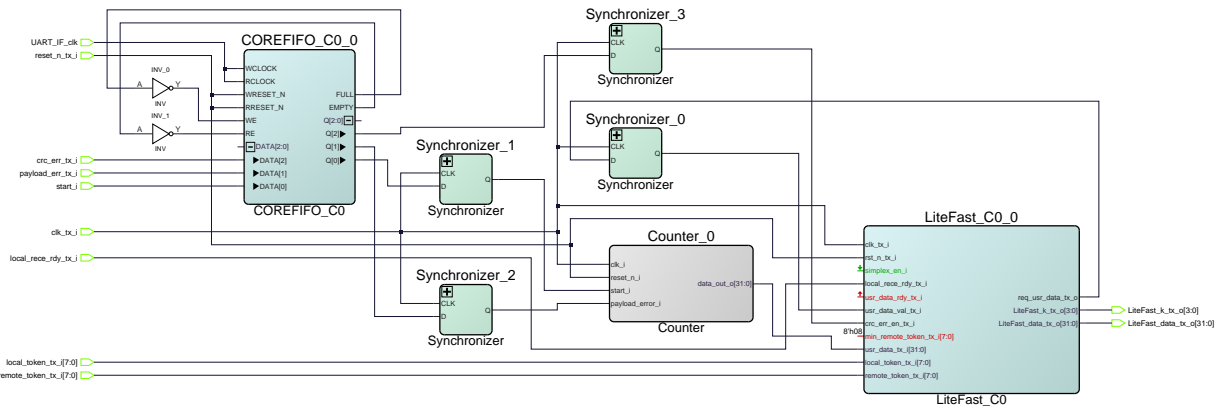
The following figure shows the sub-blocks of UART\_IF\_0 block.

Figure 1-3. UART\_IF\_0 Sub-Blocks



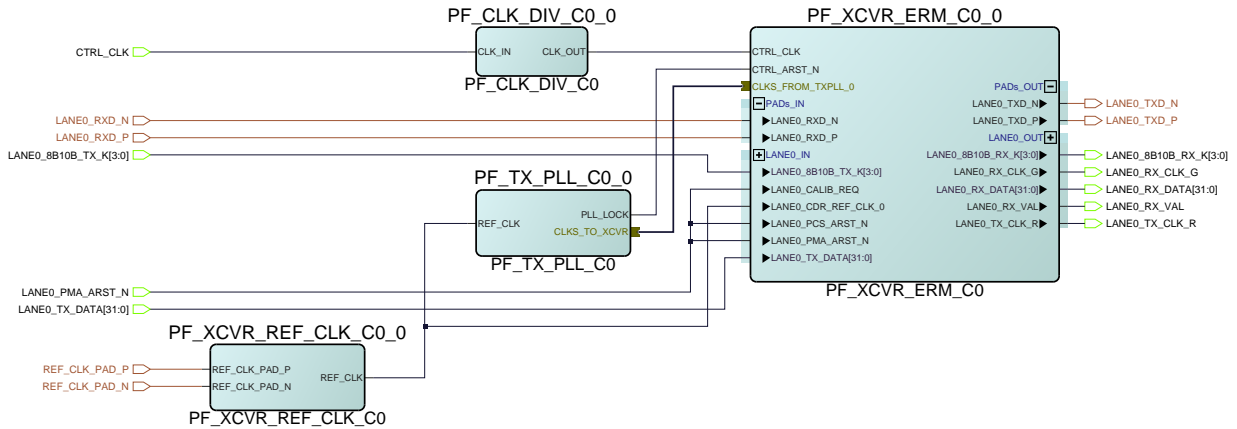
The following figure shows the sub-blocks of LiteFast\_Transmitter\_0 block.

Figure 1-4. LiteFast\_Transmitter\_0 Sub-Blocks



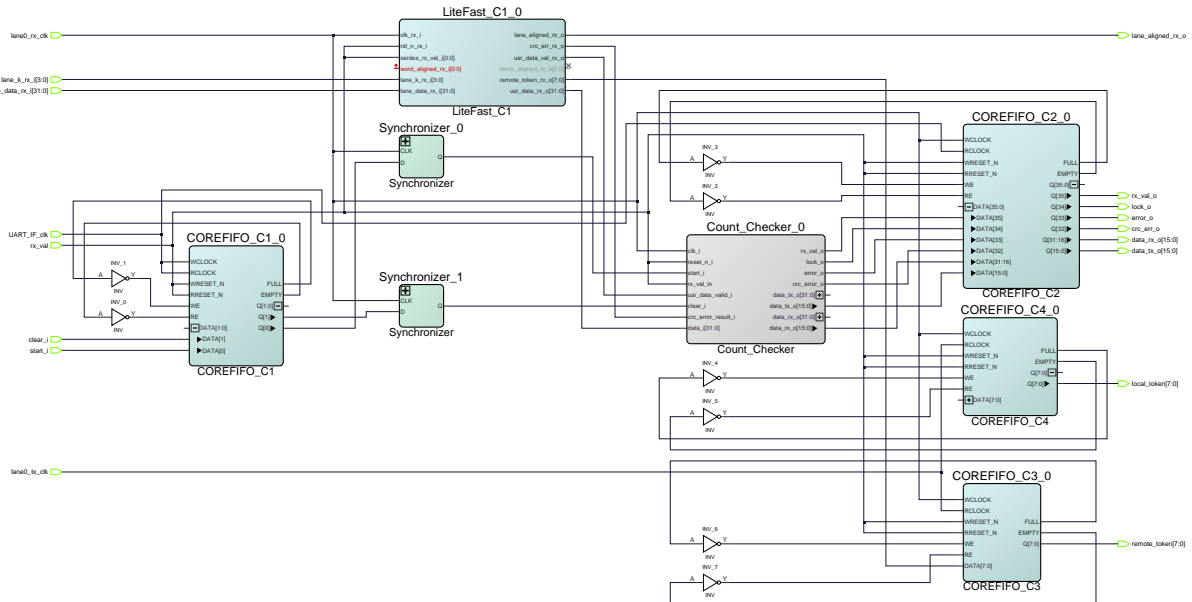
The following figure shows the sub-blocks of Transceiver\_IF\_0 block.

Figure 1-5. Transceiver\_IF\_0 Sub-Blocks



The following figure shows the sub-blocks of LiteFast\_Receiver\_0 block.

Figure 1-6. LiteFast\_Receiver\_0 Sub-Blocks



The following table lists the important I/O signals of the design.

Table 1-2. I/O Signals

Signal	Description
<b>Input Signals</b>	
PADs_IN	Transceiver LANE0_RXD_P and LANE0_RXD_N on the board
REF_CLK_PAD_P_0 and REF_CLK_PAD_N_0	This is the differential reference clock generated from the on-board 125 MHz oscillator
RX	This is the input signal received by the UART interface from the GUI
<b>Output Signals</b>	
PADs_OUT	LANE0_TXD_P and LANE0_TXD_N looped back to LANE0_RXD_P and LANE0_RXD_N using SMA cables
TX	This is the output data received by the GUI from the UART interface

### 1.3.2 IP Configuration

The following sections describe the user-defined blocks, IP blocks, and their configurations for each top-level block.

#### 1.3.2.1 UART\_IF\_0 block

The UART\_IF\_0 block contains the PF\_CCC, CoreUART, Reset\_Synchronizer\_0, and FabUART modules. These modules are described in the following sections.

##### 1.3.2.1.1 PF\_CCC\_C0\_0

The PF\_CCC\_C0\_0 block provides 125 MHz output fabric clock to the COREUART\_C0\_0 and FabUART\_0 modules, which are fabric blocks. The following figures show the input and output configurations of PF\_CCC\_C0\_0.

Figure 1-7. PF\_CCC\_C0\_0 Clock Options PLL

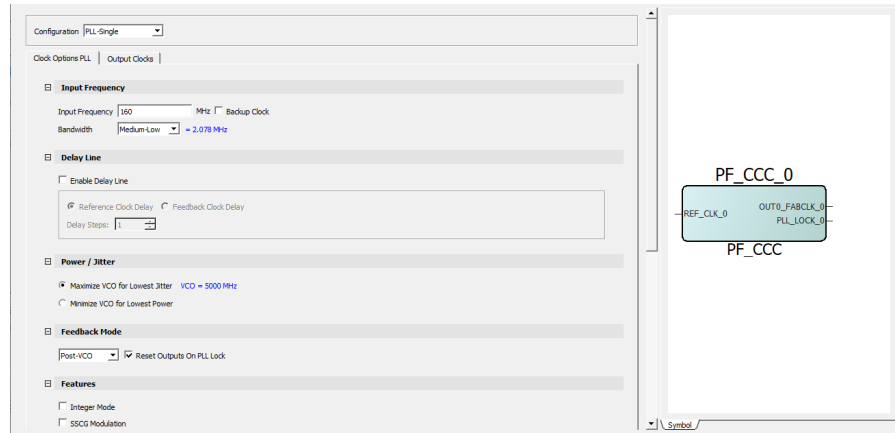
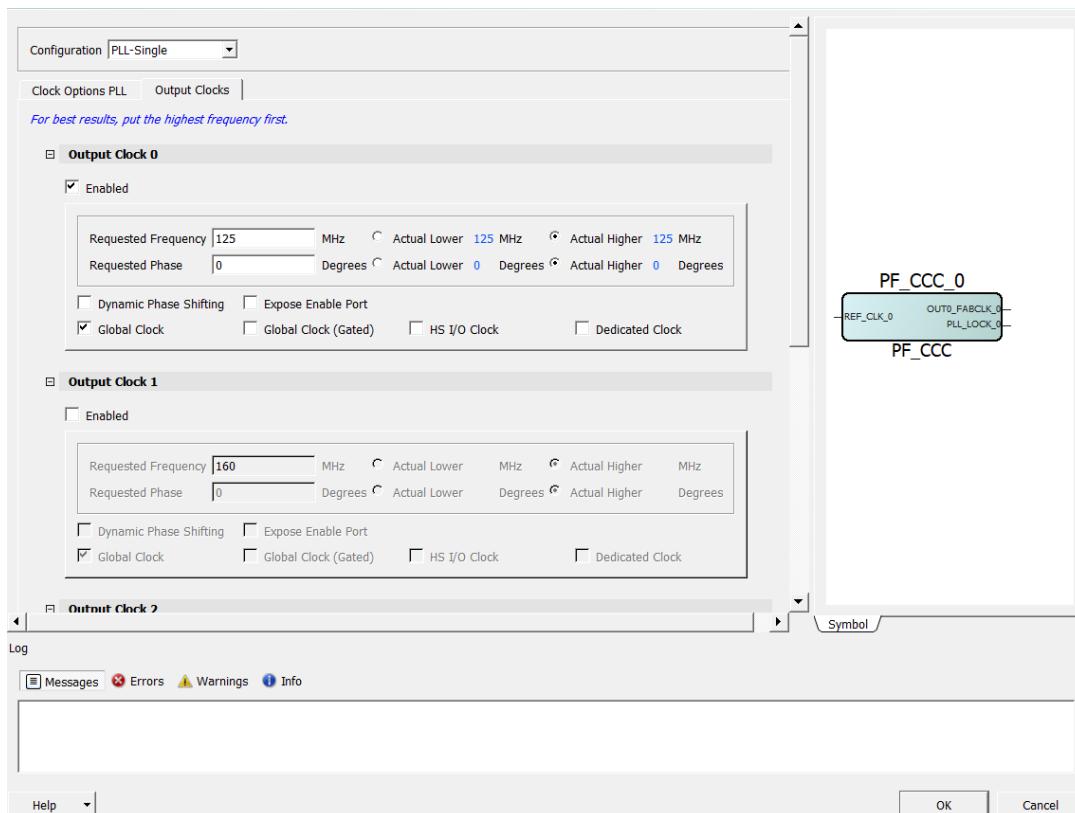


Figure 1-8. PF\_CCC\_C0\_0 Output Configuration



For more information about PF\_CCC, see [PolarFire FPGA and PolarFire SoC FPGA Clocking Resources User Guide](#).

### 1.3.2.1.2 COREUART\_C0\_0

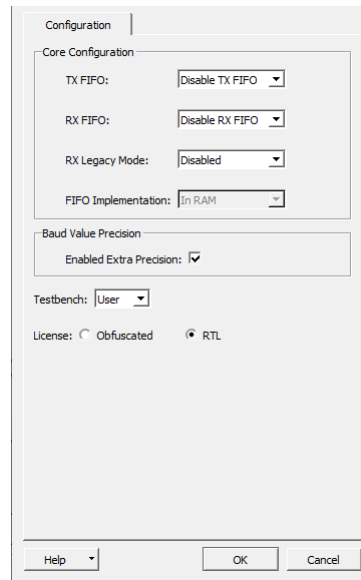
User inputs from the GUI are received by the COREUART module, which converts this serial input data to parallel data and forwards the data to the FabUART module for further processing. Both COREUART\_C0\_0 and FabUART\_0 modules run at the same frequency. Hence, the TX and RX FIFO options are disabled in the COREUART\_C0\_0 configurator as shown in [Figure 1-9](#). For more information about CoreUART, see CoreUART Handbook. This IP module can be downloaded from the Libero catalog.

### 1.3.2.1.3 Reset\_Synchronizer\_0

The Reset\_Synchronizer\_0 block is a two stage synchronizer, which synchronizes the PLL\_LOCK\_0 signal.

The following figure shows COREUART\_0 configuration.

**Figure 1-9. COREUART\_0 Configurator**



### 1.3.2.1.4 FabUART\_0

The FabUART module drives the Counter\_0 and Count\_Checker\_0 modules and receives the data and error information from the Count\_Checker\_0 module. The FabUART module passes this data to the CoreUART, which converts this parallel data to serial and forwards the data to the GUI. It is a user-defined module.

### 1.3.2.2 LiteFast\_Transmitter\_0

This block contains the Counter\_0, LiteFast\_tx\_0, Synchronizer\_0, and COREFIFO\_0 modules. These modules are described in the following sections.

#### 1.3.2.2.1 Counter\_0

The counter\_0 module implements a 32-bit counter that transmits incremental data to the LiteFast\_tx\_0 module at each clock cycle.

#### 1.3.2.2.2 LiteFast\_C0\_0

LiteFast\_C0\_0 is the instantiation of the LiteFast IP configured as transmitter. This IP core receives the incremental data from Counter\_0 and converts that data to LiteFast frames. These frames are sent to the Transceiver\_IF\_0 block over a single lane. The following figure shows this data width and lane configuration.

Figure 1-10. LiteFast\_C0\_0 Configurator

Configuration

g\_DATA\_WID: 32

g\_LANE\_NUM: 1

LiteFast\_Mode: Transmitter Only

License: Obfuscated

Help OK Cancel

### 1.3.2.2.3 Synchronizer

The Synchronizer block is a two stage synchronizer, which synchronizes the clock domain crossing signals.

### 1.3.2.2.4 COREFIFO\_C0\_0

The COREFIFO\_C0\_0 IP is used for clock domain crossing of the `crc_err_en_tx_i`, `payload_error_i`, and `start_i` signals from `UART_IF_CLK` domain to `LANE0_TX_CLK`. The following figure shows the COREFIFO\_C0\_0 configuration. For more information about CoreFIFO, see CoreFIFO Handbook. This IP can be downloaded from the Libero catalog.

Figure 1-11. COREFIFO\_C0\_0 Configurator

Configuration

FIFO Operation

Controller Type: With 64x12 uSRAM

Memory Pipeline: Pipelined

ECC: Disabled

Reset Type: Asynchronous Reset

Synchronizer Stages: 2

Clocks: Dual Clock

Optimized for: High Speed

Prefetch (Single Clock Cycle Read)  FWFT (First-Word Fall-Through)

Clock, Enable and Reset

Read Enable: Active High

Write Enable: Active High

Read Port

Width: 3

Depth: 32

Write Port

Width: 3

Depth: 32

Data Handshake

Read Data Valid  Write Acknowledgement

Disable Reads when FIFO is Empty  Disable Writes when FIFO is Full

Flags

Almost EMPTY Almost EMPTY Threshold: 4

Almost FULL Almost FULL Threshold: 60

Underflow  Overflow

Write Count  Read Count

Testbench: User

License: RTL

Check Configuration OK Cancel

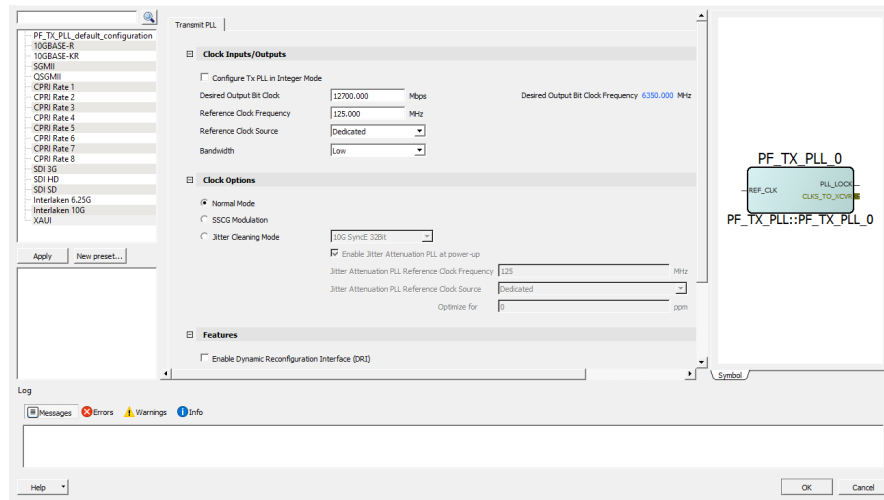
### 1.3.2.3 Transceiver\_IF\_0

This block contains the `PF_TX_PLL_C0_0`, `PF_XCVR_REF_CLK_C0_0`, and the `PF_XCVR_C0_0` modules. These modules are described in the following sections.

#### 1.3.2.3.1 PF\_TX\_PLL\_C0\_0

The `PF_TX_PLL` IP block provides the reference clock to the transceiver lane. This block is configured as shown in the following figure.

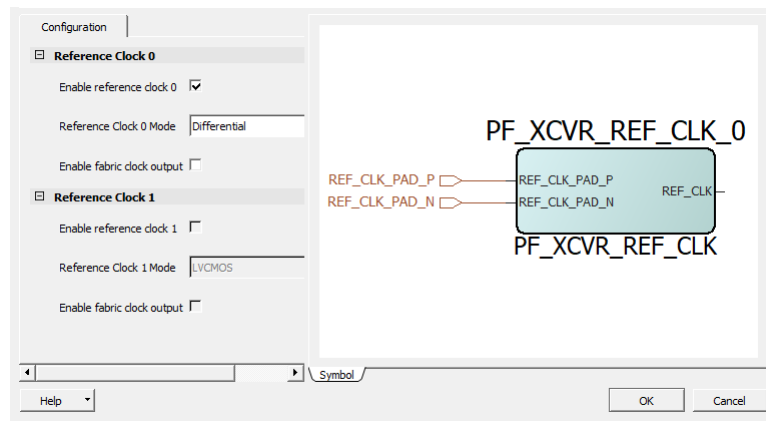
Figure 1-12. PF\_TX\_PLL Configurator



### 1.3.2.3.2 PF\_XCVR\_REF\_CLK\_C0\_0

The PF\_XCVR\_REF\_CLK IP block provides the reference clock to PF\_TX\_PLL and CDR PLL. This block is configured as shown in the following figure.

Figure 1-13. PF\_XCVR\_REF\_CLK Configurator



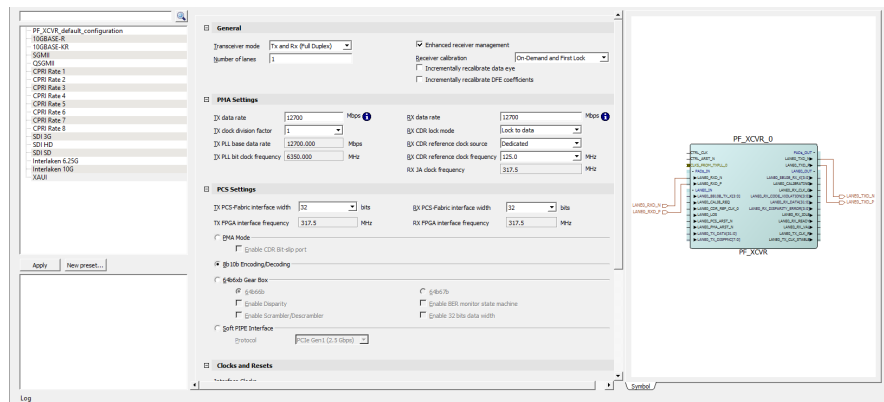
### 1.3.2.3.3 PF\_XCVR\_C0\_0

The PolarFire high-speed transceiver (PF XCVR) is a hard IP block that supports high-speed data rates ranging from 250 Mbps to 12.7 Gbps. In this demo, the transceiver block (PF\_XCVR) is configured in 8b10b mode on lane 0 with a CDR reference clock of 125 MHz to support 12.7 Gbps data rate.

PolarFire Transmit PLL (PF\_TX\_PLL) is used to send the reference clock feed to the transceiver. The dedicated reference clock (PF\_XCVR\_REF\_CLK) drives the PF\_TX\_PLL to generate the desired output clock for the 12.7 Gbps data rate. For more information about the PolarFire Transceiver, see [PolarFire FPGA and PolarFire SoC FPGA Transceiver User Guide](#).

The following figure shows the PF\_XCVR\_C0\_0 configuration.

Figure 1-14. PF\_XCVR\_0 Configurator

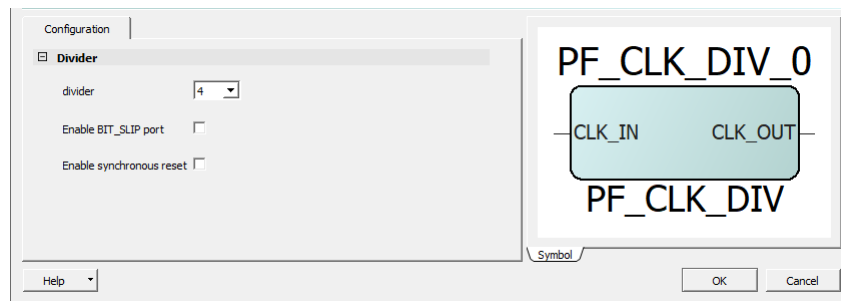


#### 1.3.2.3.4 PF\_CLK\_DIV\_C0\_0

This block divides the input clock from PF\_OSC\_C0\_0 by a factor of 4 and provides 40 MHz clock, which is used as cntrl\_clk to the PF\_XCVR\_C0\_0.

The following figure shows the PF\_CLK\_DIV\_C0\_0 configuration.

Figure 1-15. PF\_CLK\_DIV\_C0\_0 Configurator



#### 1.3.2.4 LiteFast\_Receiver\_0

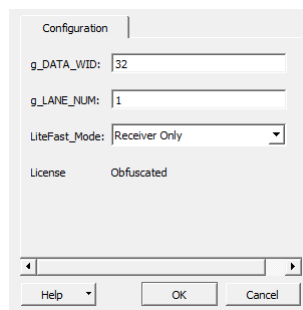
This block contains the LiteFast\_C1\_0, COREFIFO, and Count\_Checker\_0 modules. These modules are described in the following sections.

##### 1.3.2.4.1 LiteFast\_C1\_0

LiteFast\_C1\_0 is the instantiation of the LiteFast IP configured as receiver. This block recognizes the LiteFast frame, and extracts the user data (payload) from that frame and forwards it to the Count\_Checker\_0 block.

The following figure shows the LiteFast\_C1\_0 configuration.

Figure 1-16. LiteFast\_C1\_0 Configurator



##### 1.3.2.4.2 COREFIFO

The LiteFast\_Receiver block contains the following instances of COREFIFO:

- COREFIFO\_C3\_0 stores the storage capacity of the LiteFast\_C1\_0 block instantiated on the receiving board, and sends this data to the LiteFast\_C0\_0 block on the transferring board for preventing excess transfer of frames.
- COREFIFO\_C4\_0 stores the storage capacity of the LiteFast\_C1\_0 block instantiated on the transferring board for preventing excess transfer of frames. The following figure shows the COREFIFO\_C3\_0 configuration.

**Figure 1-17. COREFIFO\_C3\_0 Configurator**

The following figure shows the COREFIFO\_C4\_0 configuration.

**Figure 1-18. COREFIFO\_C4\_0 Configurator**

- COREFIFO\_C1\_0 is used for clock domain crossing of the clear\_i and start\_i signals from UART\_IF\_CLK domain to LANE0\_RX\_CLK. The following figure shows the COREFIFO\_C1\_0 configuration.

**Figure 1-19. COREFIFO\_C1\_0 Configurator**

The screenshot shows the configuration window for COREFIFO\_C1\_0. The 'FIFO Operation' section includes: Controller Type: With 64x12 USRAM, Clocks: Dual Clock, Memory Pipeline: Pipelined, Synchronizer Stages: 2, ECC: Disabled, Reset Type: Asynchronous Reset, and Optimized for: High Speed. There are checkboxes for Prefetch (Single Clock Cycle Read) and FWFT (First-Word Fall-Through), both of which are unchecked. The 'Clock, Enable and Reset' section has Read Enable and Write Enable both set to Active High. The 'Read Port' section has Width: 2 and Depth: 32. The 'Write Port' section has Width: 2 and Depth: 32. The 'Data Handshake' section has checkboxes for Read Data Valid, Write Acknowledgement, Disable Reads when FIFO is Empty (checked), and Disable Writes when FIFO is Full (checked). The 'Flags' section has checkboxes for Almost EMPTY, Almost FULL, Underflow, Overflow, Write Count, and Read Count. Almost EMPTY Threshold is 4 and Almost FULL Threshold is 60. The Testbench is set to User. Buttons at the bottom include Check Configuration, OK, and Cancel.

- COREFIFO\_C2\_0 is used for clock domain crossing of the data\_tx\_o bus, data\_rx\_o bus, crc\_error\_o, error\_o, lock\_o, and rx\_val\_o signals from LANE0\_RX\_CLK domain to UART\_IF\_CLK. The following figure shows the COREFIFO\_C2\_0 configuration.

**Figure 1-20. COREFIFO\_C2\_0 Configurator**

The screenshot shows the configuration window for COREFIFO\_C2\_0. The 'FIFO Operation' section includes: Controller Type: With 10x20 LSRAM, Clocks: Dual Clock, Memory Pipeline: Pipelined, Synchronizer Stages: 2, ECC: Disabled, Reset Type: Asynchronous Reset, and Optimized for: High Speed. There are checkboxes for Prefetch (Single Clock Cycle Read) and FWFT (First-Word Fall-Through), both of which are unchecked. The 'Clock, Enable and Reset' section has Read Enable and Write Enable both set to Active High. The 'Read Port' section has Width: 36 and Depth: 512. The 'Write Port' section has Width: 36 and Depth: 512. The 'Data Handshake' section has checkboxes for Read Data Valid, Write Acknowledgement, Disable Reads when FIFO is Empty (checked), and Disable Writes when FIFO is Full (checked). The 'Flags' section has checkboxes for Almost EMPTY, Almost FULL, Underflow, Overflow, Write Count, and Read Count. Almost EMPTY Threshold is 4 and Almost FULL Threshold is 1020. The Testbench is set to User and the License is RTL. Buttons at the bottom include Check Configuration, OK, and Cancel.

**1.3.2.4.3 Count\_Checker\_0**

The Count\_Checker\_0 block contains a 32-bit checker that checks the incoming data with the self-generated data. An error signal is asserted whenever there is a mismatch between the estimated and captured data. It is a user-defined module.

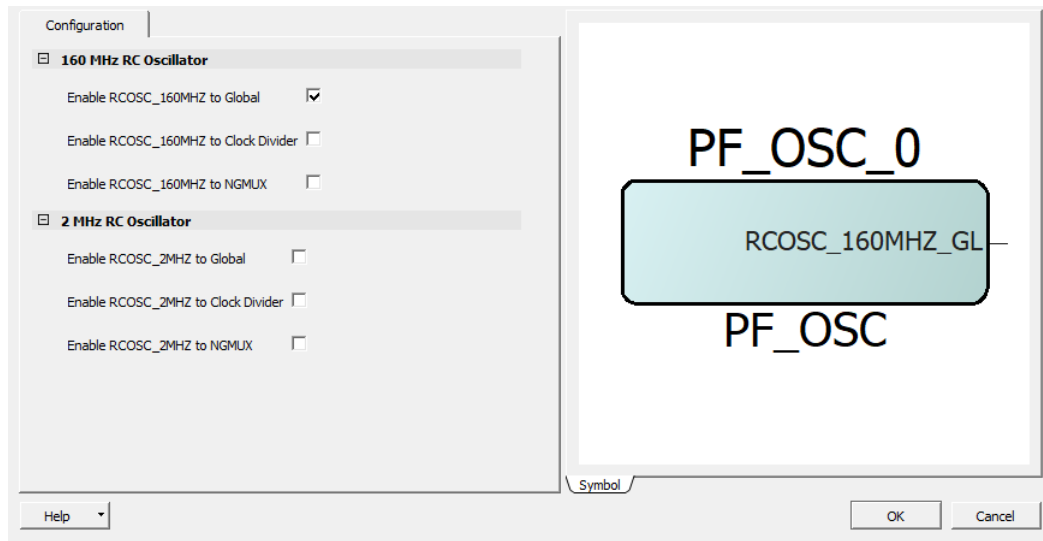
**1.3.2.5 LiteFast\_XCVR\_Top**

All the above blocks are instantiated in this module. This block also contains PF\_OSC\_C0\_0 and PF\_INIT\_MONITOR\_C0\_0. These blocks are described in the [1.3.2.5.1. PF\\_OSC\\_C0\\_0](#) and [1.3.2.5.2. PF\\_INIT\\_MONITOR\\_C0\\_0](#).

**1.3.2.5.1 PF\_OSC\_C0\_0**

This on-chip oscillator provides a clock to the PF\_CCC\_C0\_0 block in the UART\_IF and to the PF\_CLK\_DIV\_C0\_0 in the Transceiver\_IF. In the reference design, PF\_OSC\_C0\_0 is configured to provide a 160 MHz clock through a global buffer, as shown in the following figure.

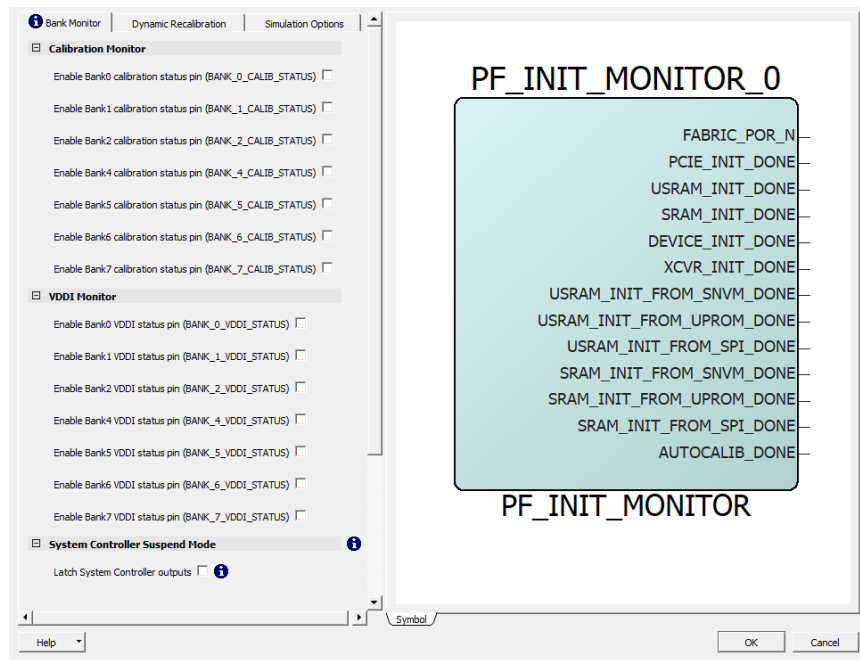
**Figure 1-21. PF\_OSC Configurator**

**1.3.2.5.2 PF\_INIT\_MONITOR\_C0\_0**

This initialization monitor asserts the DEVICE\_INIT\_DONE signal when the device initialization is done, and this signal is used to reset the transceiver.

The following figure shows polarfire initialization monitor configuration.

Figure 1-22. PolarFire Initialization Monitor Configurator

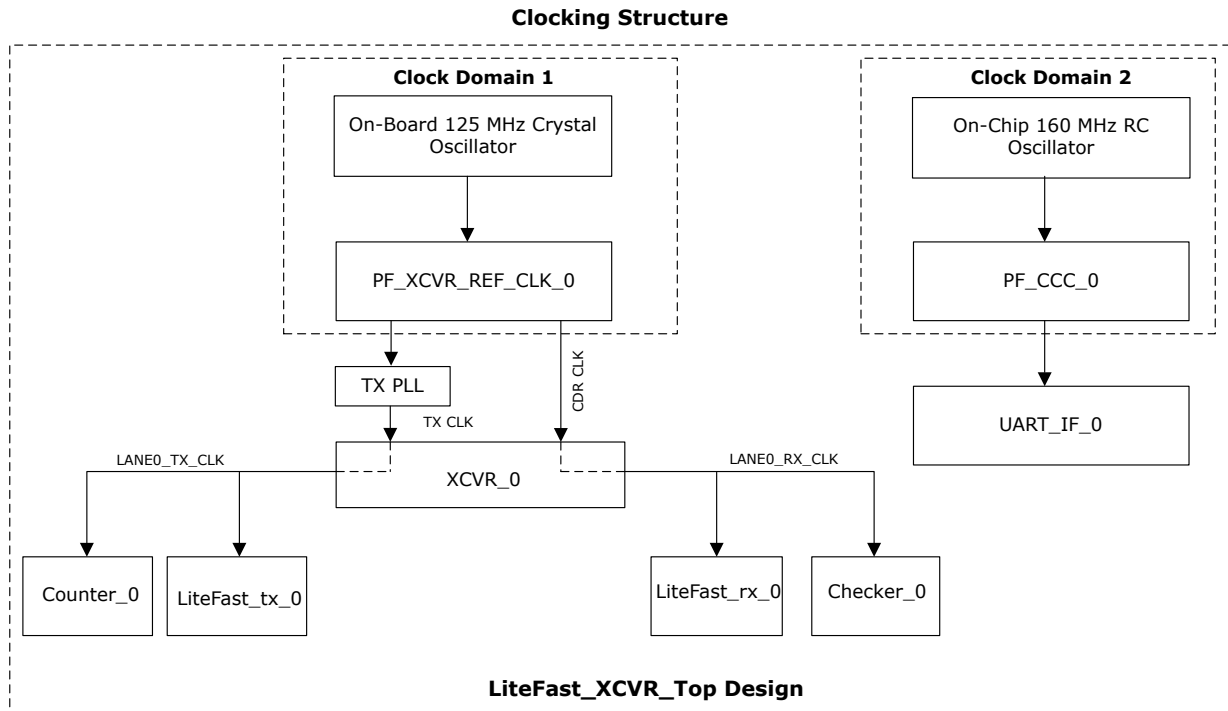


## 1.4 Clocking Structure

In the reference design, there are two clock domains. The on-board 125 MHz crystal oscillator drives the XCVR reference clock, which provides clock source to the Counter\_0, LiteFast\_tx\_0, XCVR\_0, LiteFast\_rx\_0, and Checker\_0 blocks. The on-chip 160 MHz RC oscillator drives the UART\_IF\_0 block.

The following figure shows the clocking structure in the reference design.

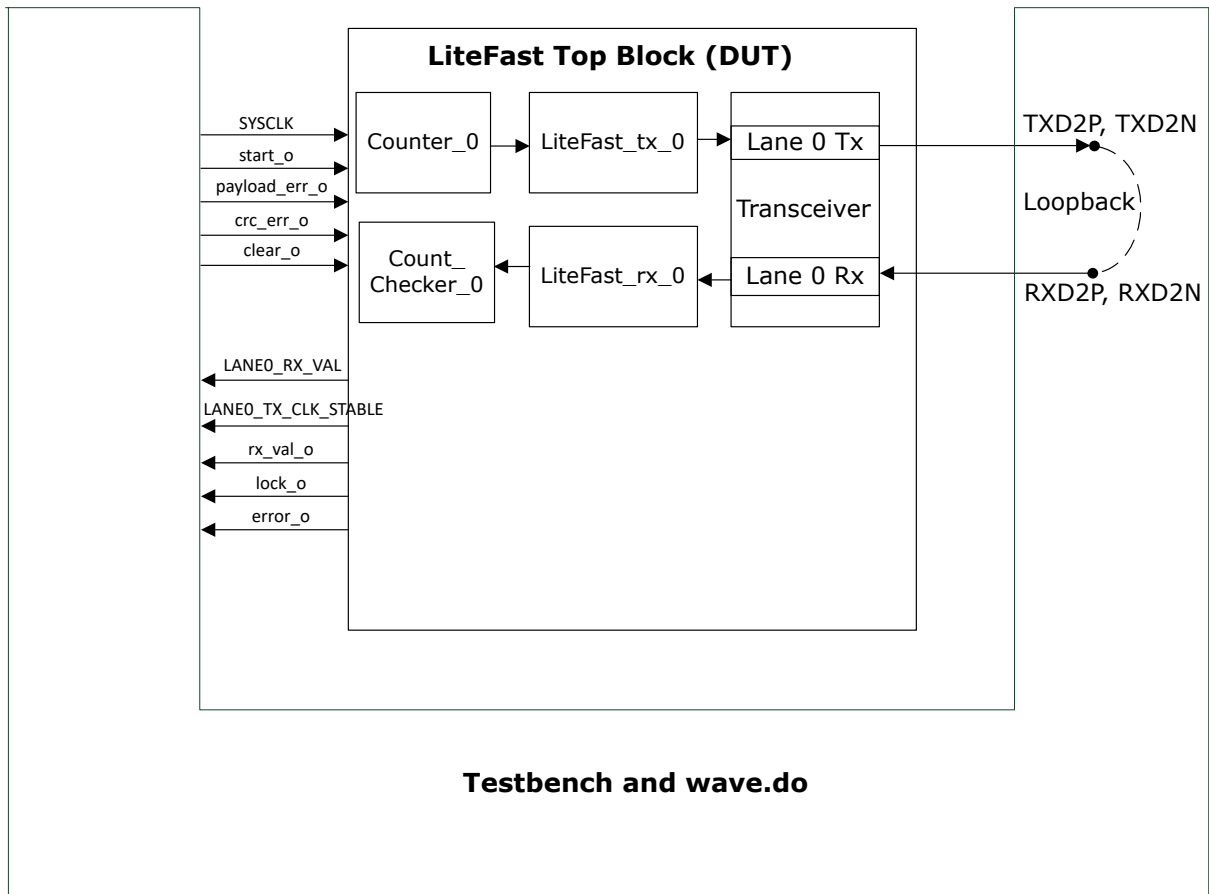
**Figure 1-23. Clocking Structure**



## 1.5 Simulating the Design

The following figure shows the interaction between testbench and the design.

Figure 1-24. Testbench and LiteFast Reference Design Interaction



The following table lists the simulation signals.

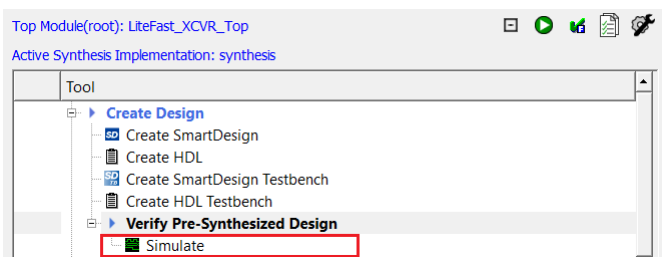
Table 1-3. Simulation Signals

Signal	Description
<b>From Testbench to DUT</b>	
SYSCLK	125 MHz system clock
start_o	This signal starts the Counter and Checker
payload_err_o	This signal injects payload error in the Counter generated data
crc_err_o	This signal injects CRC error in the LiteFast frames generated by the LiteFast_tx_0 module
clear_o	This signal disables payload error and CRC error
<b>From DUT to Testbench</b>	
LANE0_RX_VAL	This output signal of the PF_XCVR indicates that the XCVR has received and validated the LiteFast idle frames
LANE0_TX_CLK_STABLE	This output signal of the PF_XCVR indicates that the LANE0_TX_CLK is locked to transmitter frequency (TX PLL)
LANE0_TX_DATA	This data bus transmits LiteFast frames to Transceiver
LANE0_RX_DATA	This data bus receives LiteFast frames from Transceiver

.....continued	
Signal	Description
rx_val_o	This is the registered version of LANE0_RX_VAL signal in LANE0_RX_CLK domain
lock_o	This signal asserts high when the counter and checker data match
error_o	This signal asserts high when the counter and checker data mismatch
data_tx_o	This output bus of the LiteFast_rx_0 contains the de-framed data passed to the Count_Checker_0
data_rx_o	This is the Count_Checker_0 generated data for comparing with the data_tx_o

In the **Design Flow** tab, click **Simulate** under **Verify Pre-Synthesized Design** and select **open interactively** to simulate the design. The Simulate option is highlighted in the following figure.

**Figure 1-25. Simulating the Design**



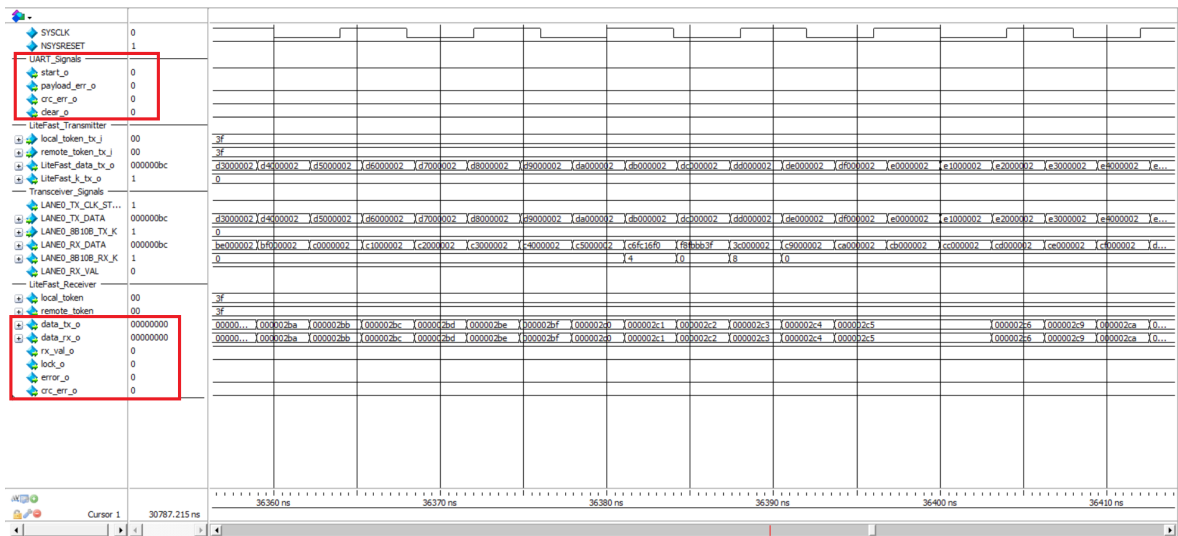
When the Simulation is initiated, ModelSim compiles all the design source files, testbench, and the stimulus, and launches the waveform window to show the simulation signals.

### 1.5.1 Simulation Flow

The following steps describe the LiteFast testbench simulation flow:

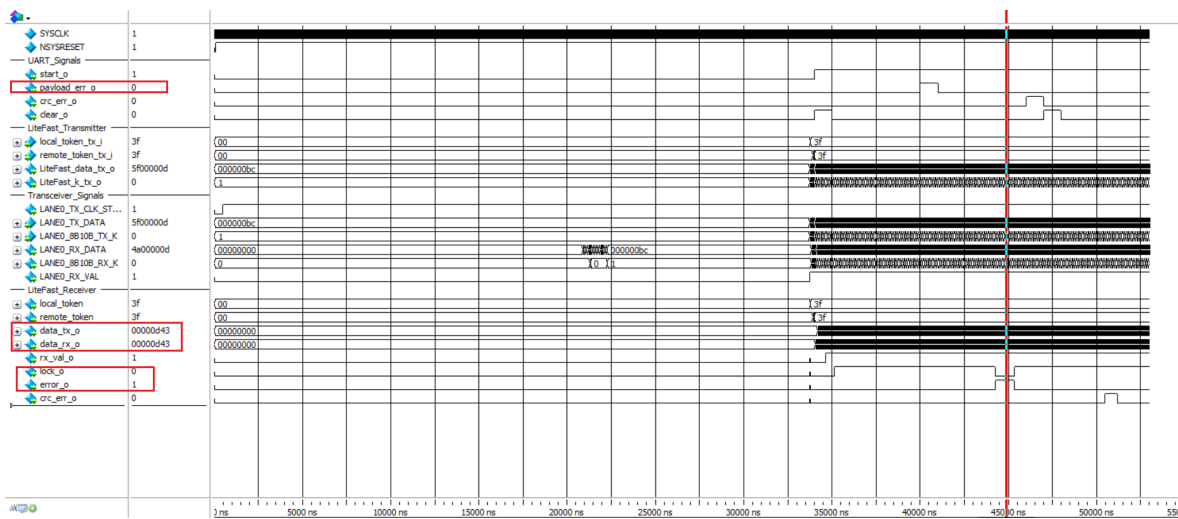
1. At 0 ns, the testbench drives the 125 MHz system clock to the DUT.
2. At ~452 ns, the testbench asserts the LANE0\_TX\_CLK\_STABLE signal. This indicates that PF\_TX\_PLL in the DUT has locked to 317.5 MHz.
3. The LiteFast IP sends the IDLE character (K28.5) to XCVR.
4. The K28.5 character (0x000000BC) is looped back from TX lane to RX lane.
5. The receiver PLL locks (LANE0\_RX\_CLK\_R) to 317.5 MHz and then, the K28.5 character asserts the LANE0\_RX\_VAL signal high at ~33741 ns.
6. The Counter\_0, LiteFast\_tx\_0, LiteFast\_rx\_0, and Count\_Checker\_0 come out of reset because of the LANE0\_RX\_VAL assertion.
7. At 34000 ns, the testbench drives the start\_o (UART\_IF\_0 output) signal high.
8. The Counter\_0 starts sending incremental data starting from 0 to LiteFast\_tx\_0, which sends that data to PF\_XCVR\_0.
9. The LiteFast\_rx\_0 receives the data from the external loopback, and the Count\_Checker\_0 compares the received data. The following figure shows the simulation waveform with no errors.

Figure 1-26. Simulation Waveform with No Errors



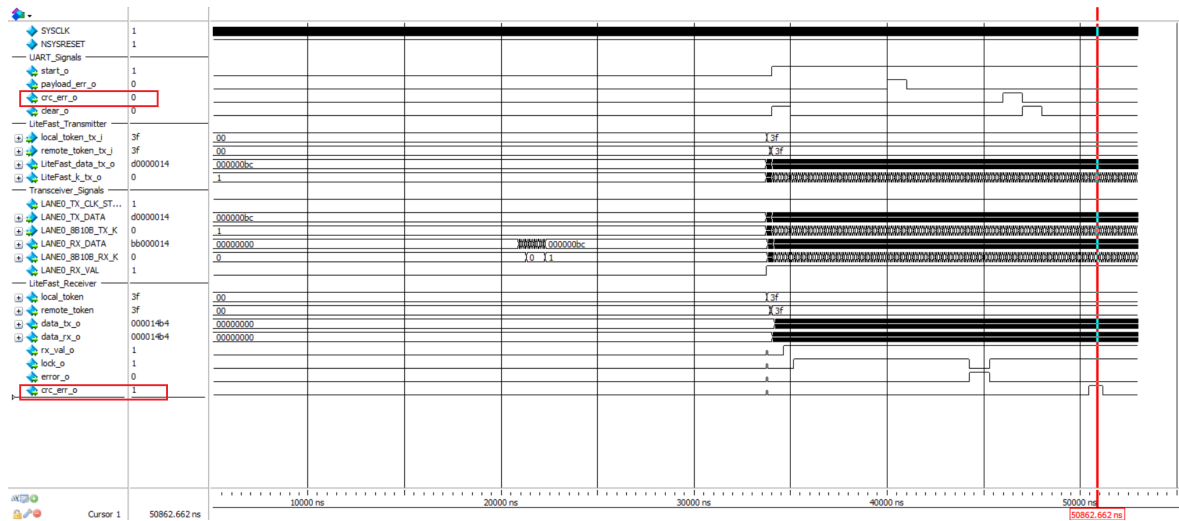
- At 40000 ns, the testbench drives the payload\_err\_o high. As a result, lock\_o goes low and error\_o goes high, which indicates that the Counter\_0 generated data does not match the Count\_Checker\_0 generated data. The following figure shows these signals.

Figure 1-27. Simulation Waveform With Payload Error



- At 41000 ns, the testbench drives the payload\_err\_o signal low.
- At 46000 ns, the testbench drives the crc\_err\_o high for simulating the CRC error. As a result, the crc\_error\_o signal goes high, which indicates the presence of CRC checksum error. The following figure shows these signals.

Figure 1-28. Simulation Waveform With CRC Error



- At 48000 ns, the testbench drives the `clear_o` signal low and the `clear` pulse high, which clears all the errors.



**Important:** The `data_rx_o`, `data_tx_o`, `rx_val_o`, `lock_o`, `error_o`, and `clear_o` signals are uninitialized when `COREFIFO_3` is in reset. As a result, these uninitialized states appear as glitches in the waveform. This behavior does not impact the design.

This concludes the testbench simulation flow of the LiteFast reference design.

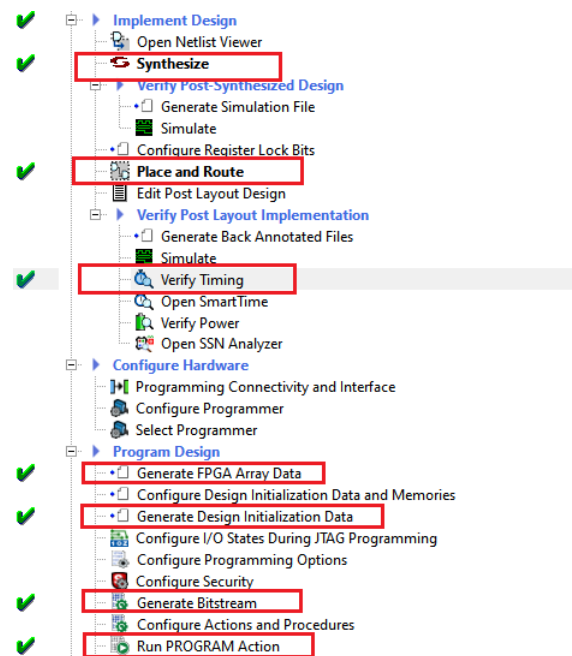
## 2. Libero Design Flow

This section describes the Libero design flow, which involves the following steps:

- Synthesize
- Place and Route
- Verify timing
- Generate FPGA Array Data
- Design and Memory Initialization
- Generate Bitstream
- Run PROGRAM Action

The following figure shows these options in the **Design Flow** tab.

Figure 2-1. Libero Design Flow Options



### 2.1 Synthesize

To synthesize the design, perform the following steps:

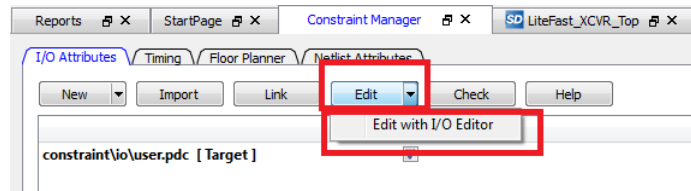
1. Double click **Synthesize** from the **Design Flow** tab. When the synthesis is successful, a green tick mark appears, as shown in [Figure 2-1](#).
2. Right click **Synthesize** and select **View Report** to view the synthesis report and log files in the **Reports** tab. We recommend viewing the `LiteFast_XCVR_Top.srr` and the `LiteFast_XCVR_Top_compile_netlist.log` files for debugging synthesis and compile errors.

### 2.2 Place and Route

To place and route the design, perform the following steps:

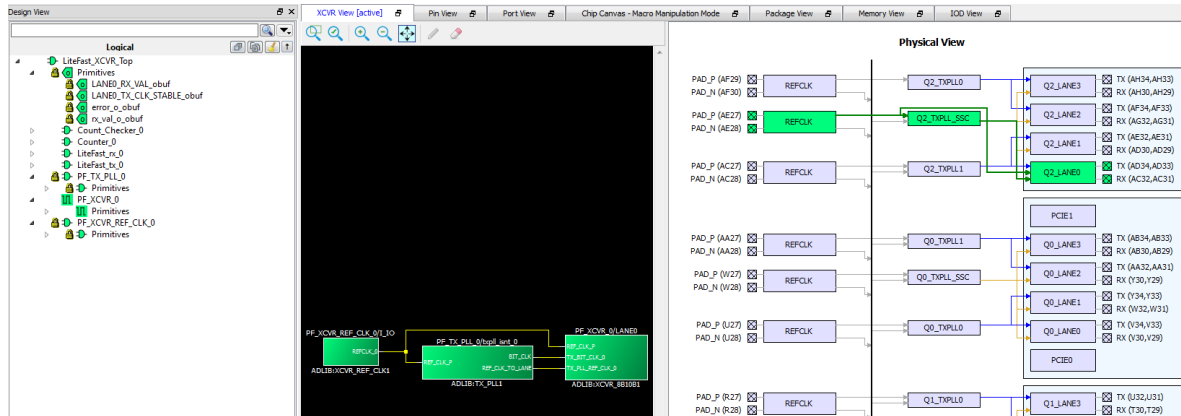
1. From Design Flow, double click **Manage Constraints** and click the **Edit with I/O Editor** option from the **I/O Attributes** tab, see the following figure.

**Figure 2-2. Starting I/O Editor**



- Place TX\_PLL, XCVR\_REF\_CLK, and PF\_XCVR TX and RX lane using the **I/O Editor**, see the following figure.

**Figure 2-3. I/O Editor XCVR View**



- Double click **Place and Route** from the **Design Flow** tab. When place and route is successful, a green tick appears, as shown in [Figure 2-1](#).
- Right click **Place and Route** and select **View Report** to view the place and route report and log files in the **Reports** tab. We recommend viewing the `LiteFast_XCVR_Top_place_and_route_constraint_coverage.xml` file for place and route constraint coverage.

### 2.2.1 Resource Utilization

The resource utilization report is written to the `LiteFast_XCVR_Top_layout_log.log` file in the **Reports > LiteFast\_XCVR\_Top report > Place and Route**. The following table lists the resource utilization of the design after place and route. These values may vary slightly for different Libero runs, settings, and seed values.

**Table 2-1. Resource Utilization**

Type	Used	Total	Percentage
4LUT	2445	299544	0.82
DFF	1640	299544	0.55
I/O Register	0	1530	0.00
Logic Element	2717	299544	0.91

### 2.3 Verify Timing

To verify timing, perform the following steps:

- Double click **Verify Timing** from the **Design Flow** tab. When the design successfully meets the timing requirements, a green tick mark appears, as shown in [Figure 2-1](#).
- Right click **Verify Timing** and select **View Report** to view the verify timing report and log files in the **Reports** tab.

## 2.4 Generate FPGA Array Data

Double click **Generate FPGA Array Data** from the **Design Flow** window.

A green tick mark is displayed after the successful generation of the FPGA array data, see [Figure 2-1](#).

## 2.5 Design and Memory Initialization

To create the XCVR initialization client, double click **Generate Design Initialization Data** from the **Design Flow** window. A green tick mark appears after the successful generation of the design initialization data, see [Figure 2-1](#).

## 2.6 Generate Bitstream

To generate the bitstream:

1. Double click **Generate Bitstream** from the **Design Flow** tab.  
When the bitstream is successfully generated, a green tick mark appears, see [Figure 2-1](#).
2. Right click **Generate Bitstream** and select **View Report** to view the corresponding log file in the **Reports** tab.

## 2.7 Run PROGRAM Action

After generating the bitstream, the PolarFire device must be programmed. Follow these steps to program the PolarFire device:

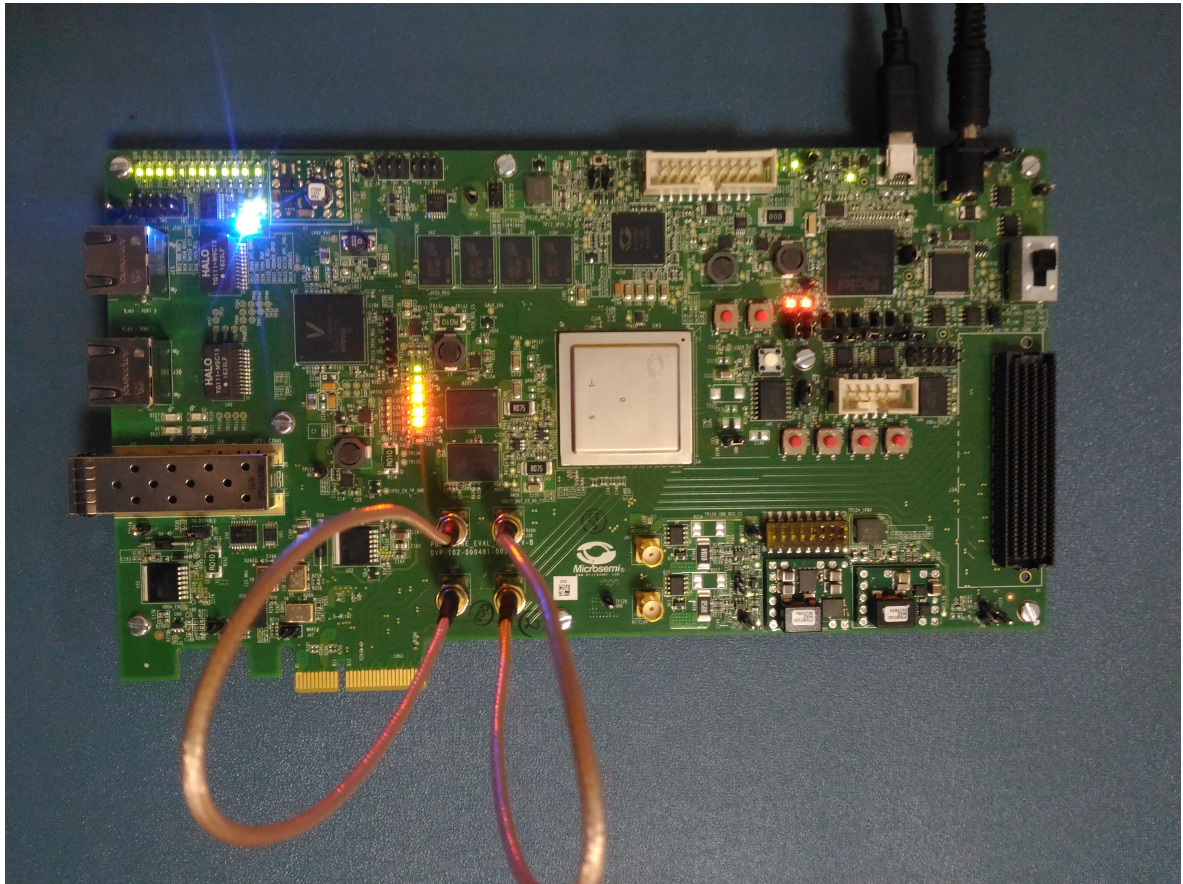
1. Ensure that the jumper settings on the board are the same as listed in the following table.

**Table 2-2. Jumper Settings**

Jumper	Description	Default
J18, J19, J20, J21, J22	Close pin 2 and 3 for programming the PolarFire® FPGA through FTDI	Closed
J28	Close pin 2 and 3 for programming through the on-board FlashPro5	Open
J26	Close pin 1 and 2 for programming through the FTDI SPI	Closed
J27	Close pin 1 and 2 for programming through the FTDI SPI	Closed
J4	Close pin 1 and 2 for manual power switching using SW3	Closed
J12	Close pin 3 and 4 for 2.5V	Closed
J46	Close pin 1 and 2 for setting the Reference Clock to 125 MHz on board oscillator	Closed

2. Connect the power supply cable to the J9 connector on the board.
3. Connect the USB cable from the Host PC to the J5 (FTDI port) on the board.
4. Power on the board using the SW3 slide switch.
5. Connect TXN to RXN and TXP to RXP using the 2 SMA-to-SMA cables as shown in the following figure.

Figure 2-4. Board Setup



6. Double click **Run PROGRAM Action** from the **Libero > Design Flow** tab.
7. Right click **Run PROGRAM Action** and select **View Report** to view the corresponding log file in the **Reports** tab.

When the device is successfully programmed, a green tick appears, as shown in [Figure 2-1](#). See [3. Running the Demo](#) to run the LiteFast standalone demo.

### 3. Running the Demo

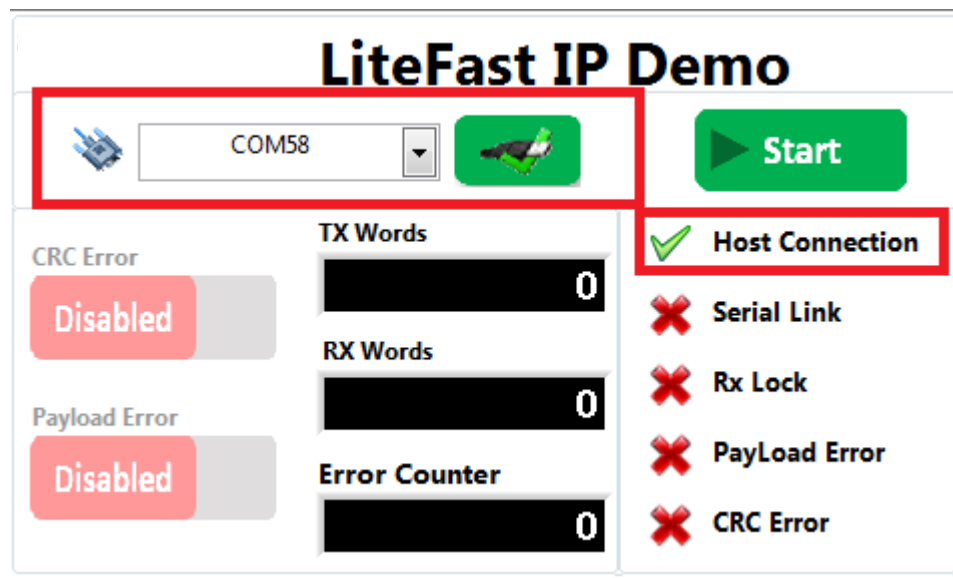
This chapter describes how to install and use the GUI to run the LiteFast demo. The following procedure assumes that:

1. The PolarFire Evaluation board is connected.
2. The PolarFire FPGA is programmed with the LiteFast design.

To run the LiteFast demo:

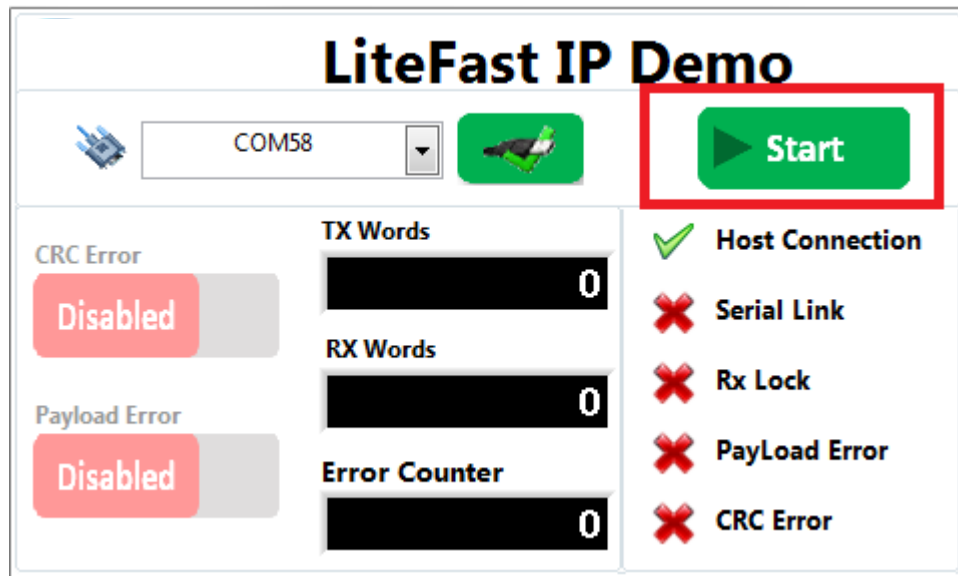
1. Extract the contents of the `mpf_an4792_v2022p2_df.zip` file.
2. From the `mpf_an4792_v2022p2_df\GUI\GUI_Installer` folder, double click the `setup.exe` file.
3. Follow the instructions displayed on the installation wizard.
4. After successful installation, `LiteFast_IP_Demo` appears on the Start menu of the host PC desktop.
5. From the Start menu, click `LiteFast_IP_Demo`.
6. Double click the `LiteFast_GUI` application from the installation directory to start the GUI application.
7. The GUI detects the COM port number and automatically connects to the board, as shown in the following figure. Port numbers may vary.  
The Host Connection status changes to a green tick mark, as shown in the following figure.

Figure 3-1. Detecting the COM Port and Host Connection Status



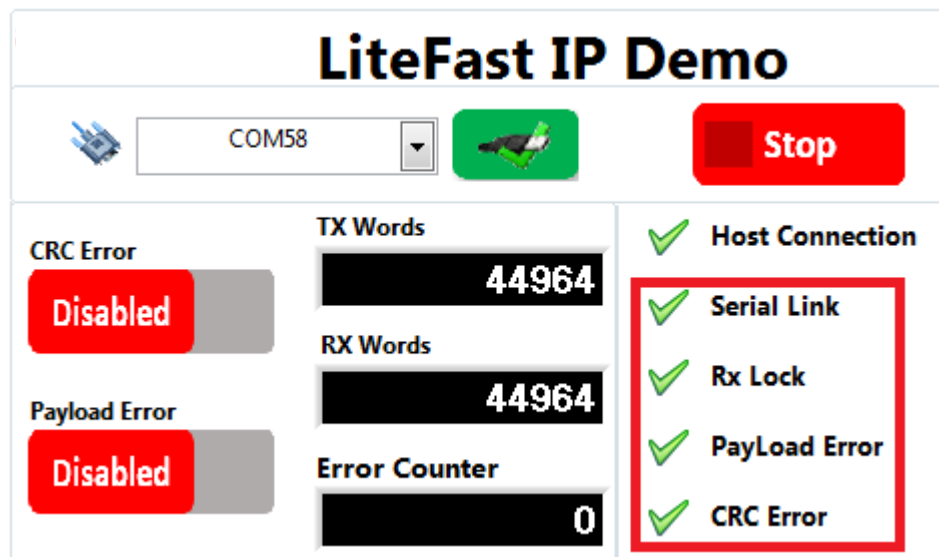
8. Click the **Start** button to start the LiteFast demo, as shown in the following figure.

Figure 3-2. Starting the LiteFast Demo



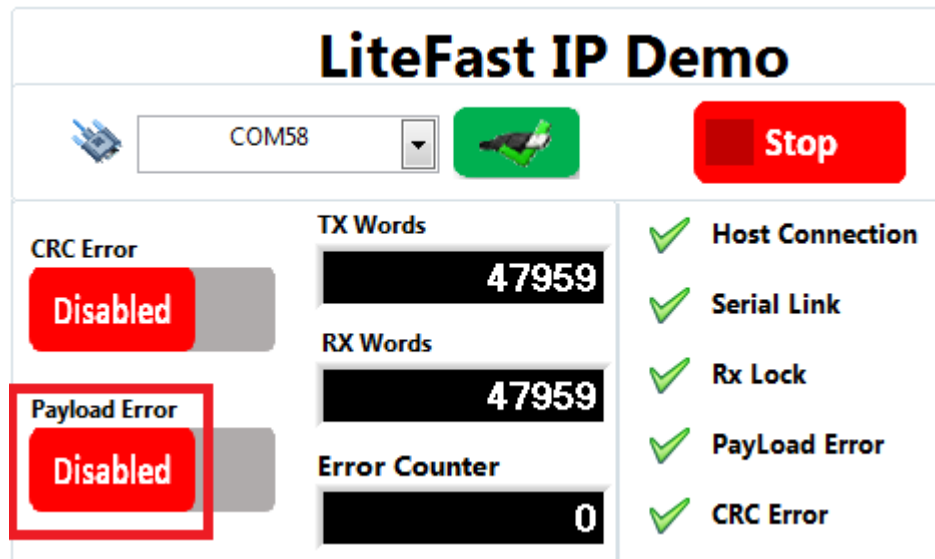
The Serial Link, Rx Lock, Payload Error, and CRC Error changes to a green tick mark, as shown in the following figure.

Figure 3-3. Overall Status



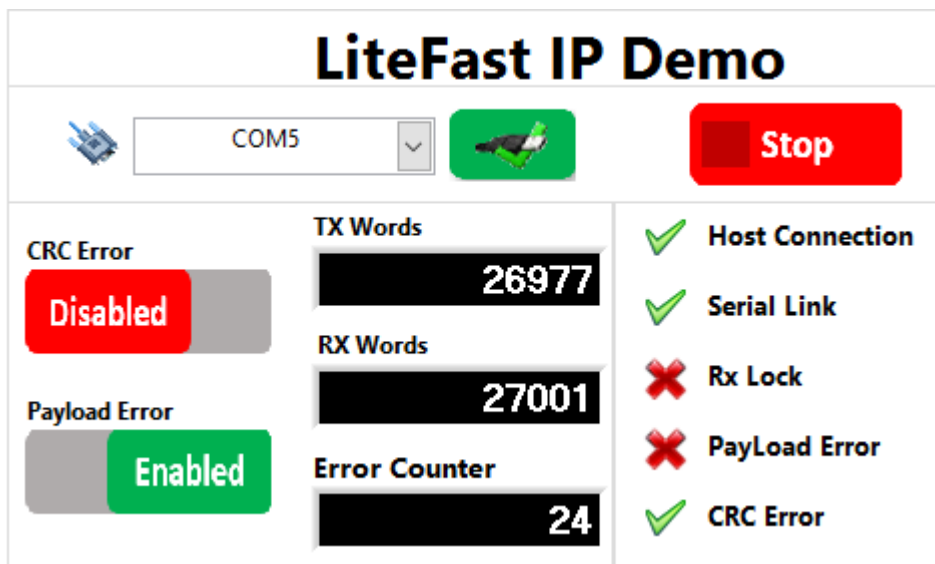
- Click the **Disabled** button in the **Payload Error** area, as shown in the following figure, to inject payload error in the Counter generated data.

Figure 3-4. Enabling Payload Error



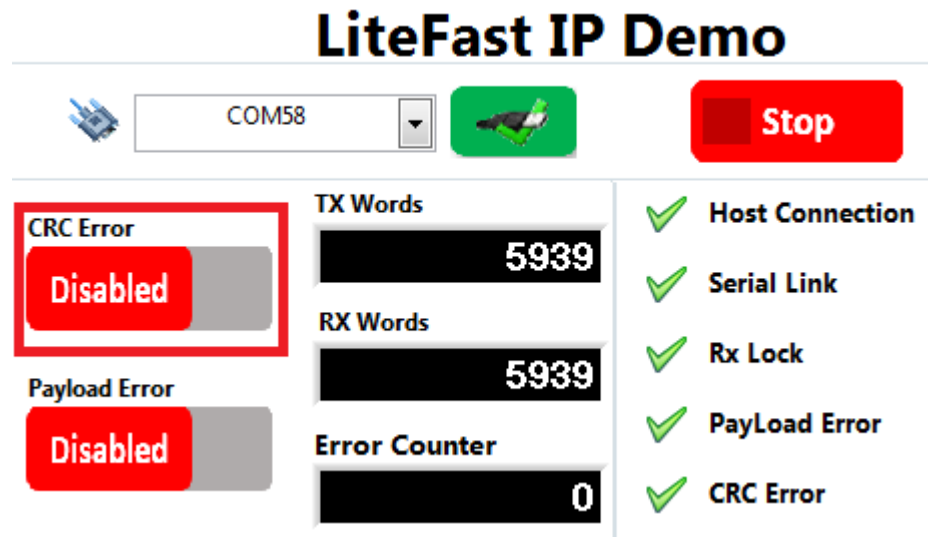
The Rx Lock and Payload Error status changes to a red cross mark, as shown in the following figure.

Figure 3-5. Checking Payload Error status



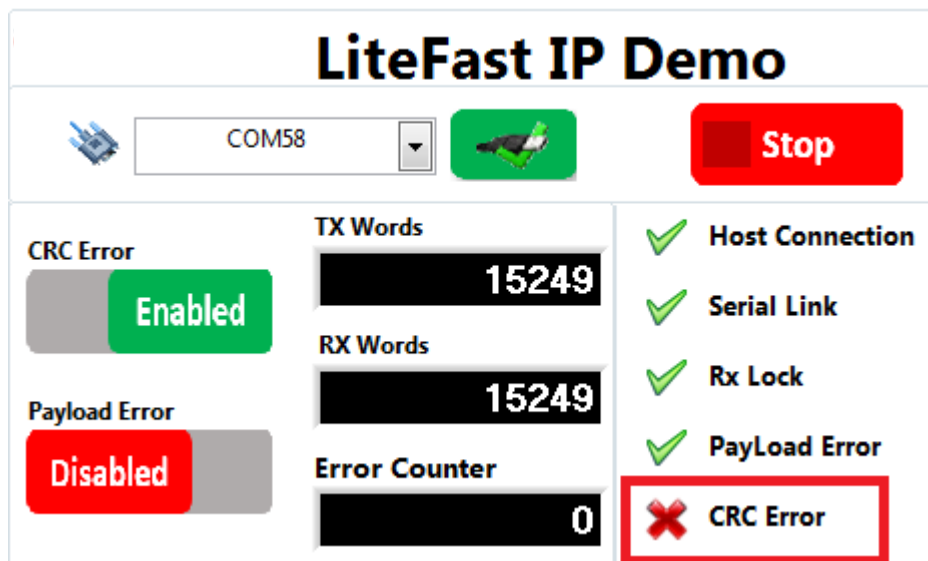
- Click **Disabled** button in the **CRC Error** area, as shown in the following figure, to inject CRC error in the Counter generated data.

Figure 3-6. Enabling CRC Error



11. The CRC Error status changes to a red cross mark, as shown in the following figure.

Figure 3-7. Checking CRC Error Status



12. **TX Words** displays the number of transmitted data words. This number rolls over after 65535 words.
13. **RX Words** displays the number of received data words. This number rolls over after 65535 words.
14. **Error Counter** displays the packet loss.



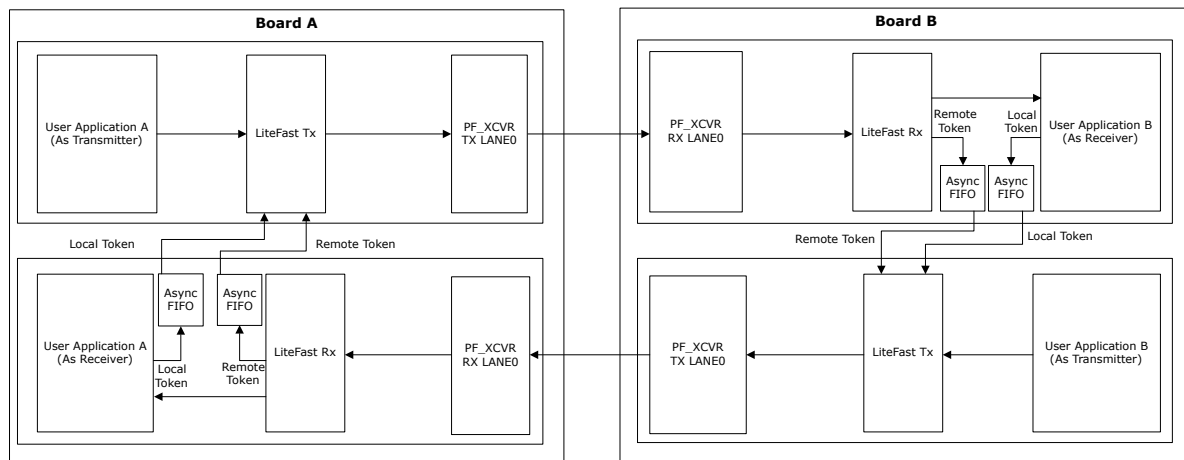
**Important:** The error counter may not exactly co-relate to TX and RX words due to the frequency gap between the LiteFast transmitter IP and the CoreUART IP.

This concludes running the LiteFast IP Demo.

## 4. Using LiteFast For Board-to-Board Data Transfer

This section describes how to implement the LiteFast IP for board-to-board data transfer. For example, there are two Microchip boards (Board A and B) running different applications (Application A and B). The following figure outlines the LiteFast transmitter and receiver designs in boards A and B for implementing a full-duplex data transfer.

**Figure 4-1. Data Transfer Between Board A and B**



**Important:** The depth of the FIFO must be defined based on the user application.

The following steps describe how to implement full-duplex data transfer between board A and B:

1. Program the reference design on both boards A and B.
2. Connect TXDP and TXDN of board A to RXDP and RXDN of board B using two SMA cables.
3. Connect TXDP and TXDN of board B to RXDP and RXDN of board A using two SMA cables.
4. Install the LiteFast GUI on two separate Host PCs (A and B).
5. Connect Host PC A to board A and Host PC B to board B using USB cables.
6. Power-up boards A and B.
7. Start the LiteFast GUI on Host PC A and B.
8. Click **Start** on both the instances of the GUI running on Host PC A and B.  
The number of TX Words displayed on the GUI running on Host PC A matches the number of RX Words displayed on the GUI running on Host PC B. Similarly, the number of TX Words displayed on the GUI running on Host PC B matches the number of RX Words displayed on the GUI running on Host PC A.

This concludes the full-duplex data transfer demo.

## 5. Appendix 1: Programming the Device Using FlashPro Express

This chapter describes how to program the PolarFire device with the .job programming file using Flashpro Express. The .job file is available at the following design files folder location: mpf\_an4792\_v2022p2\_df\Programming\_job

Follow these steps:

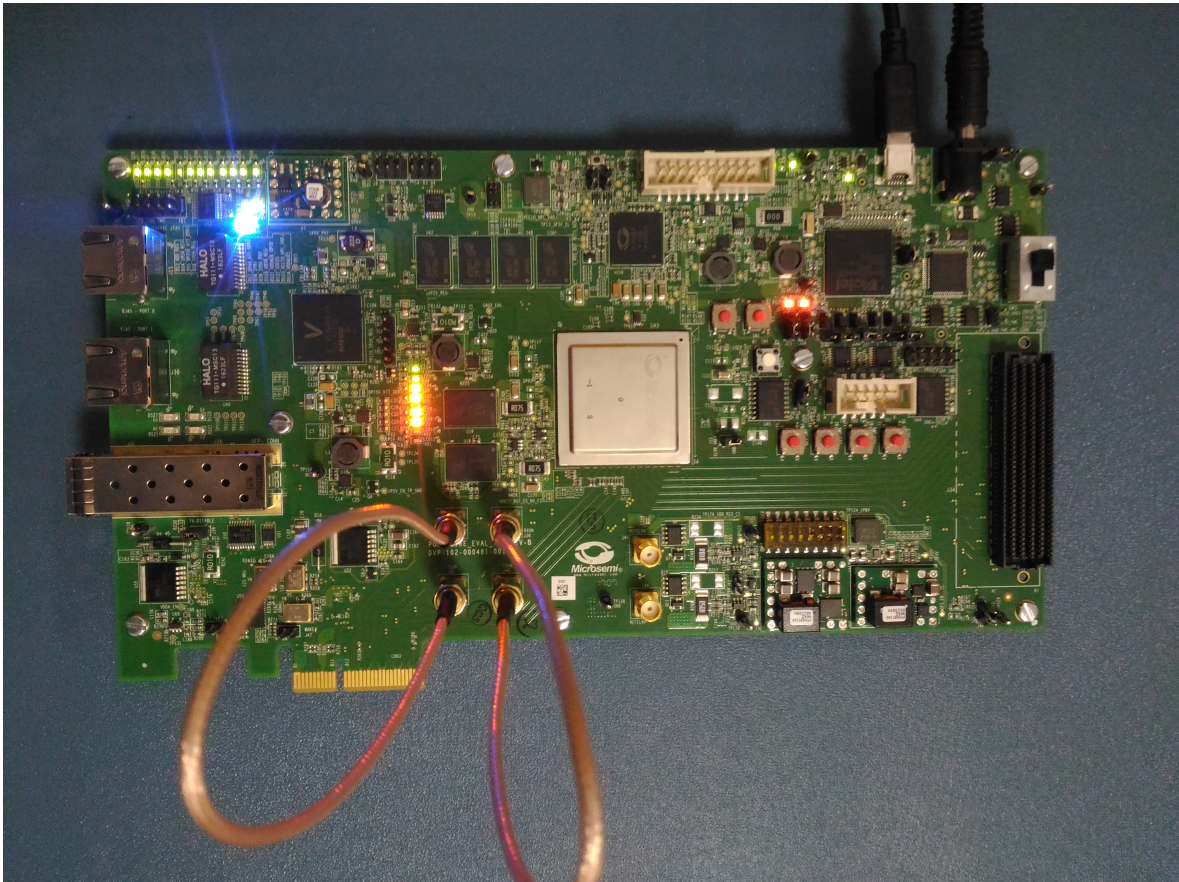
1. Ensure that the jumper settings on the board are the same as listed in the following table.

**Table 5-1. Jumper Settings**

Jumper	Description	Default
J18, J19, J20, J21, and J22	Close pin 2 and 3 for programming the PolarFire® FPGA through FTDI	Closed
J28	Close pin 2 and 3 for programming through the on-board FlashPro5	Open
J26	Close pin 1 and 2 for programming through the FTDI SPI	Closed
J27	Close pin 1 and 2 for programming through the FTDI SPI	Closed
J4	Close pin 1 and 2 for manual power switching using SW3	Closed
J12	Close pin 3 and 4 for 2.5V	Closed
J46	Close pin 1 and 2 for setting the Reference Clock to 125 MHz on board oscillator	Closed

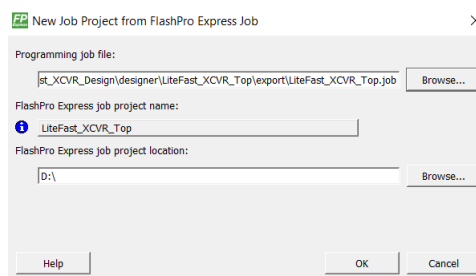
2. Connect the power supply cable to the J9 connector on the board.
3. Connect the USB cable from the Host PC to J5 (FTDI port) on the board.
4. Power on the board using the SW3 slide switch.
5. Connect **TXN** to **RXN** and **TXP** to **RXP** using the 2 SMA-to-SMA cables, see the following figure. The following figure shows the board setup.

Figure 5-1. Board Setup



6. On the host PC, start the FlashPro Express software.
7. Click **New Project** to create a new project.  
In the **New Project** window, do the following and click **OK**.
8. Browse the job file in the design files.
9. Choose the location of the FlashPro Express project.  
The following figure shows new project window.

Figure 5-2. New Project Window



10. Click **OK** to program the device.

FlashPro Express programs the .job file through the programmer type connected to the board.

## 6. Appendix 2: Running the TCL Script

TCL scripts are provided in the design files folder under the directory `TCL_Scripts`. If required, the design flow can be reproduced from Design Implementation till generation of the job file.

To run the TCL, follow the steps below:

1. Launch the Libero software.
2. Select **Project > Execute Script**.
3. Click **Browse** and select `script.tcl` from the downloaded `TCL_Scripts` directory.
4. Click **Run**.

After successful execution of the TCL script, the Libero project is created within the `TCL_Scripts` directory.

For more information about TCL scripts, refer to `mpf_an4792_v2022p2_df/TCL_Scripts/readme.txt`.

Refer to [Tcl Commands Reference Guide](#) for more details on TCL commands. Contact Microchip Technical Support for any queries encountered while running the TCL script.

## 7. Appendix 3: References

This section lists documents that provide more information about the LiteFast standard and IP cores used in the reference design.

- For more information about LiteFast IP, see LiteFast IP User Guide. This IP module can be downloaded from the Libero catalog.
- For more information about PF\_CCC, see [PolarFire FPGA and PolarFire SoC FPGA Clocking Resources User Guide](#).
- For more information about CoreFIFO, see CoreFIFO Handbook. This IP module can be downloaded from the Libero catalog.
- For more information about CoreUART, see CoreUART Handbook. This IP module can be downloaded from the Libero catalog.
- For information about PolarFire transceiver blocks, PF\_TX\_PLL, and PF\_XCVR\_REF\_CLK, see [PolarFire FPGA and PolarFire SoC FPGA Transceiver User Guide](#)
- For more information about Libero, ModelSim, and Synplify, see the [Libero SoC Documentation](#) web page.

## 8. 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.

**Table 8-1. Revision History**

Revision	Date	Descriptions
A	12/2022	The following is the list of changes in the revision A of the document: <ul style="list-style-type: none"> <li>• Migrated the document from Microsemi template to Microchip template</li> <li>• The document number was updated to DS00004792 from 50200783</li> <li>• The document ID was updated to AN4792 from DG0783</li> <li>• Updated the document for Libero v2022.2</li> </ul>
4.0	—	Updated the document for Libero v2022.2
3.0	—	The following is a summary of the changes in revision 3.0 of this document: <ul style="list-style-type: none"> <li>• Updated the document for Libero v2021.2</li> <li>• Updated <a href="#">Table 1-1</a>, <a href="#">Table 1-2</a>, and <a href="#">Table 2-1</a></li> </ul>
2.0	—	A note about an inconsistent 125 MHz oscillator that does not supply 125 MHz constantly on few PolarFire Evaluation boards was added, see <a href="#">Table 1-2</a>
1.0	—	The first publication of this document

---

## Microchip FPGA Support

---

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at [www.microchip.com/support](http://www.microchip.com/support). Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

---

## Microchip Information

---

---

### The Microchip Website

---

Microchip provides online support via our website at [www.microchip.com/](http://www.microchip.com/). This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

---

### Product Change Notification Service

---

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to [www.microchip.com/pcn](http://www.microchip.com/pcn) and follow the registration instructions.

---

## Customer Support

---

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: [www.microchip.com/support](http://www.microchip.com/support)

---

## Microchip Devices Code Protection Feature

---

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

## Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at [www.microchip.com/en-us/support/design-help/client-support-services](http://www.microchip.com/en-us/support/design-help/client-support-services).

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

## Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, KoD, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICTail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-

ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2022, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved.

ISBN: 978-1-6683-1705-1

## Quality Management System

---

For information regarding Microchip's Quality Management Systems, please visit [www.microchip.com/quality](http://www.microchip.com/quality).

## Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
<p><b>Corporate Office</b> 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: <a href="http://www.microchip.com/support">www.microchip.com/support</a> Web Address: <a href="http://www.microchip.com">www.microchip.com</a></p> <p><b>Atlanta</b> Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455</p> <p><b>Austin, TX</b> Tel: 512-257-3370</p> <p><b>Boston</b> Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088</p> <p><b>Chicago</b> Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075</p> <p><b>Dallas</b> Addison, TX Tel: 972-818-7423 Fax: 972-818-2924</p> <p><b>Detroit</b> Novi, MI Tel: 248-848-4000</p> <p><b>Houston, TX</b> Tel: 281-894-5983</p> <p><b>Indianapolis</b> Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380</p> <p><b>Los Angeles</b> Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800</p> <p><b>Raleigh, NC</b> Tel: 919-844-7510</p> <p><b>New York, NY</b> Tel: 631-435-6000</p> <p><b>San Jose, CA</b> Tel: 408-735-9110 Tel: 408-436-4270</p> <p><b>Canada - Toronto</b> Tel: 905-695-1980 Fax: 905-695-2078</p>	<p><b>Australia - Sydney</b> Tel: 61-2-9868-6733</p> <p><b>China - Beijing</b> Tel: 86-10-8569-7000</p> <p><b>China - Chengdu</b> Tel: 86-28-8665-5511</p> <p><b>China - Chongqing</b> Tel: 86-23-8980-9588</p> <p><b>China - Dongguan</b> Tel: 86-769-8702-9880</p> <p><b>China - Guangzhou</b> Tel: 86-20-8755-8029</p> <p><b>China - Hangzhou</b> Tel: 86-571-8792-8115</p> <p><b>China - Hong Kong SAR</b> Tel: 852-2943-5100</p> <p><b>China - Nanjing</b> Tel: 86-25-8473-2460</p> <p><b>China - Qingdao</b> Tel: 86-532-8502-7355</p> <p><b>China - Shanghai</b> Tel: 86-21-3326-8000</p> <p><b>China - Shenyang</b> Tel: 86-24-2334-2829</p> <p><b>China - Shenzhen</b> Tel: 86-755-8864-2200</p> <p><b>China - Suzhou</b> Tel: 86-186-6233-1526</p> <p><b>China - Wuhan</b> Tel: 86-27-5980-5300</p> <p><b>China - Xian</b> Tel: 86-29-8833-7252</p> <p><b>China - Xiamen</b> Tel: 86-592-2388138</p> <p><b>China - Zhuhai</b> Tel: 86-756-3210040</p>	<p><b>India - Bangalore</b> Tel: 91-80-3090-4444</p> <p><b>India - New Delhi</b> Tel: 91-11-4160-8631</p> <p><b>India - Pune</b> Tel: 91-20-4121-0141</p> <p><b>Japan - Osaka</b> Tel: 81-6-6152-7160</p> <p><b>Japan - Tokyo</b> Tel: 81-3-6880-3770</p> <p><b>Korea - Daegu</b> Tel: 82-53-744-4301</p> <p><b>Korea - Seoul</b> Tel: 82-2-554-7200</p> <p><b>Malaysia - Kuala Lumpur</b> Tel: 60-3-7651-7906</p> <p><b>Malaysia - Penang</b> Tel: 60-4-227-8870</p> <p><b>Philippines - Manila</b> Tel: 63-2-634-9065</p> <p><b>Singapore</b> Tel: 65-6334-8870</p> <p><b>Taiwan - Hsin Chu</b> Tel: 886-3-577-8366</p> <p><b>Taiwan - Kaohsiung</b> Tel: 886-7-213-7830</p> <p><b>Taiwan - Taipei</b> Tel: 886-2-2508-8600</p> <p><b>Thailand - Bangkok</b> Tel: 66-2-694-1351</p> <p><b>Vietnam - Ho Chi Minh</b> Tel: 84-28-5448-2100</p>	<p><b>Austria - Wels</b> Tel: 43-7242-2244-39 Fax: 43-7242-2244-393</p> <p><b>Denmark - Copenhagen</b> Tel: 45-4485-5910 Fax: 45-4485-2829</p> <p><b>Finland - Espoo</b> Tel: 358-9-4520-820</p> <p><b>France - Paris</b> Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79</p> <p><b>Germany - Garching</b> Tel: 49-8931-9700</p> <p><b>Germany - Haan</b> Tel: 49-2129-3766400</p> <p><b>Germany - Heilbronn</b> Tel: 49-7131-72400</p> <p><b>Germany - Karlsruhe</b> Tel: 49-721-625370</p> <p><b>Germany - Munich</b> Tel: 49-89-627-144-0 Fax: 49-89-627-144-44</p> <p><b>Germany - Rosenheim</b> Tel: 49-8031-354-560</p> <p><b>Israel - Ra'anana</b> Tel: 972-9-744-7705</p> <p><b>Italy - Milan</b> Tel: 39-0331-742611 Fax: 39-0331-466781</p> <p><b>Italy - Padova</b> Tel: 39-049-7625286</p> <p><b>Netherlands - Drunen</b> Tel: 31-416-690399 Fax: 31-416-690340</p> <p><b>Norway - Trondheim</b> Tel: 47-72884388</p> <p><b>Poland - Warsaw</b> Tel: 48-22-3325737</p> <p><b>Romania - Bucharest</b> Tel: 40-21-407-87-50</p> <p><b>Spain - Madrid</b> Tel: 34-91-708-08-90 Fax: 34-91-708-08-91</p> <p><b>Sweden - Gothenberg</b> Tel: 46-31-704-60-40</p> <p><b>Sweden - Stockholm</b> Tel: 46-8-5090-4654</p> <p><b>UK - Wokingham</b> Tel: 44-118-921-5800 Fax: 44-118-921-5820</p>