PolarFire FPGA Splash Kit JESD204B Standalone Interface Application Note

AN5978

Introduction (Ask a Question)

This document describes how to run the JESD204B standalone demo design on the PolarFire® Splash Board using the JESD204B Standalone Demo GUI application. The GUI application is packaged along with the design files. The demo design is a reference design built using the PolarFire high-speed transceiver blocks and the CoreJESD204BTX and CoreJESD204BRX IP cores. It operates in Loopback mode by sending the CoreJESD204BTX data to the CoreJESD204BRX IP core through the transceiver lanes, which are looped back on the board. This loopback setup facilitates a standalone JESD interface demo that does not require Analog-to-Digital Converter (ADC) or Digital-to-Analog Converters (DAC).

Microchip PolarFire devices have embedded, high-speed transceiver blocks that can handle data rates ranging from 250 Mbps to 12.5 Gbps. The transceiver (PF_XCVR) module integrates several functional blocks to support multiple high-speed serial protocols within the FPGA. JESD204B is a high-speed serial interface standard for data converters developed by the JEDEC committee. The JESD204B standard reduces the number of data inputs and outputs between the high-speed data converters and receivers.

Microchip provides CoreJESD204BTX and CoreJESD204BRX IP cores that implement the transmitter and receiver interfaces of the JESD204B standard. These IP cores are easy to integrate with JESD204B- based data converters to develop high-bandwidth applications such as wireless infrastructure transceivers, software-defined radios, medical imaging systems, and radar and secure communications. These IP cores support link widths from x1 to x4, and link rates from 250 Mbps to 12.5 Gbps per lane using subclass 0, 1 and 2.

For more information about the JESD204B interface design implementation, and all the necessary blocks and IP cores instantiated in Libero® SoC, see Demo Design.

The JESD204B standalone interface design can be programmed using any of the following options:

- Using the `.job` file: To program the device using the `.job` file provided along with the design files, see Programming the Device Using FlashPro Express.
- Using Libero SoC: To program the device using Libero SoC, see Running the Demo Design. Use this option when the demo design is modified.

Table of Contents

1. Design Requirements (Ask a Question)

The following table lists the resources required to run the demo.

Table 1-1. Design Requirements

Requirement	Version
Operating System	Windows® 10 and 11
Hardware	
PolarFire® Splash Kit with MPF300T-1FCG484E device	Rev 2 or later
Software	
FlashPro Express	For all the software versions needed to create this reference design, see `readme.txt` file provided in the design files.
GUI executable (provided with the design files)	
Libero® SoC	

2. Prerequisites (Ask a Question)

Before you start, perform the following steps:

- Download and install Libero® SoC (as indicated in the website for this design) on the host PC from Libero SoC Documentation.
- Download the demo design files from www.microchip.com/en-us/application-notes/an5978.
- Install the GUI application by running the `setup.exe` file available in the design files folder: `<$Design_Files_Directory>/mpf_an5978_df/GUI`
  At the end of the installation, you may be prompted to download and install the FPGA_GUI_Pack, if it is not already available on your system.
- Alternatively, you can manually download and install the Microchip FPGA_GUI_Pack.

Important: A Libero® Gold license is required to evaluate your designs using the PolarFire® Splash Kit.

3. Demo Design (Ask a Question)

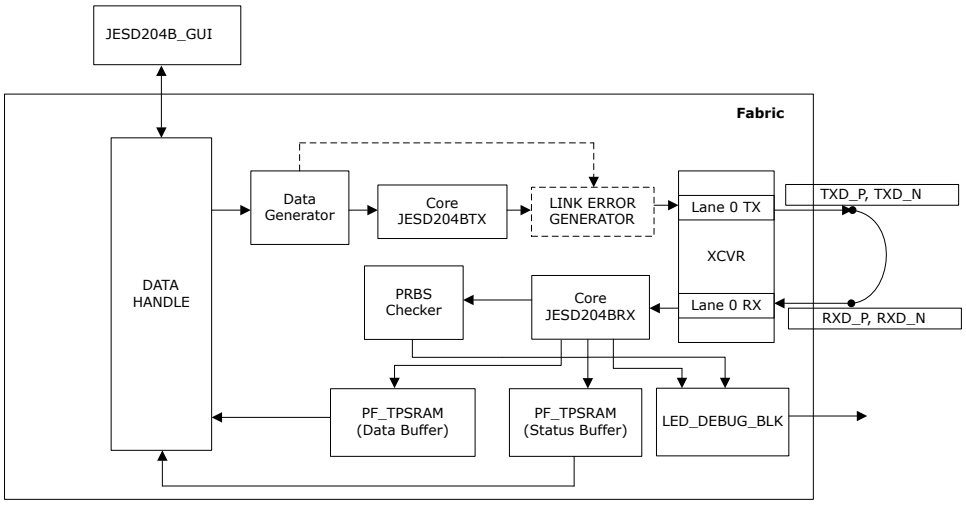
The PolarFire® JESD204B demo design is developed to interface JESD204B-compliant data converters with PolarFire devices. The design functions as follows:

1. The DATA_HANDLE_0 block interfaces with the GUI. The GUI enables the selection of either PRBS or waveform input.
2. The DATA_HANDLE_0 block forwards the input selection to the DATA_GENERATOR_0 block, which generates and sends the corresponding input data to the CoreJESD204BTX IP core.
3. The CoreJESD204BTX IP core performs JESD204B transmitter functions based on the configuration and transmits the data to the PF_XCVR (transceiver) IP core.
4. The encoded data is received by the CoreJESD204BRX IP core because the TX and RX lanes of the PF_XCVR block are looped back.
5. The CoreJESD204BRX IP core performs JESD204B receiver functions based on the configuration and transmits the data to the GUI for viewing the selected input.

Important: When a data error or link error is selected on the GUI, the error generator block generates that error and displays it on the GUI.

The following figure shows the hardware implementation of the JESD204B interface demo.

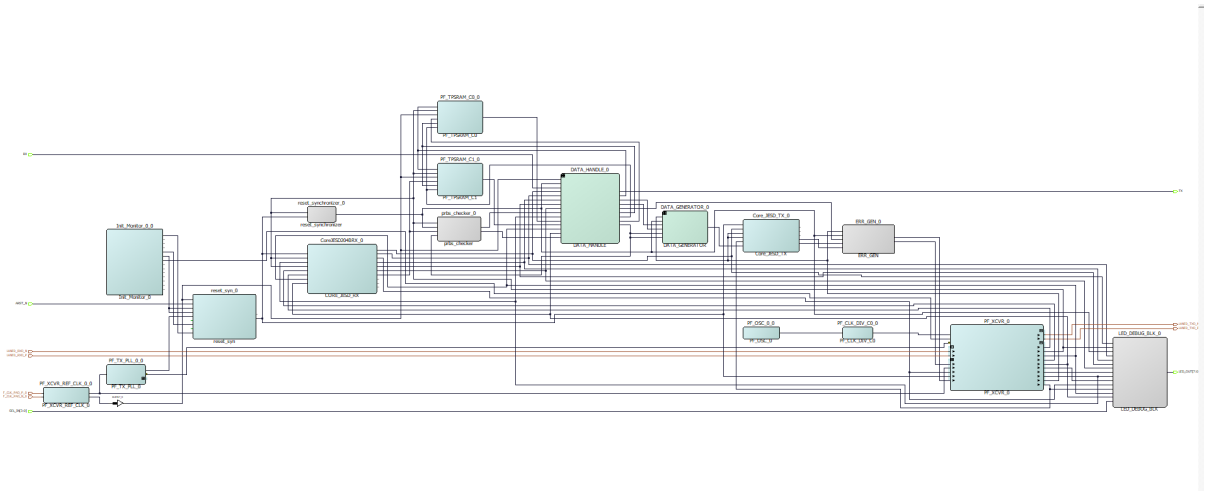
Figure 3-1. Hardware Implementation Block Diagram



3.1. Design Implementation (Ask a Question)

The following figure shows the Libero® design implementation of the JESD204B interface demo.

Figure 3-2. JESD204B Interface Design



The following table lists the important input and output signals of the design.

Table 3-1. Input and Output Signals

Signal	Description
Input Signals	
LANE0_RXD_P and LANE0_RXD_N	Transceiver receiver differential inputs
ARST_N	External reset obtained from push button switch on board
RX	Receiver of UART interface
REF_CLK_PAD_P_0 and REF_CLK_PAD_N_0	Differential reference clock obtained from the on-board 125 MHz oscillator
SEL_IN[3:0]	Signal mapped to DIPs 1, 2, 3 and 4 of SW8 dip slide switch used to debug the status and errors
Output Signals	
LANE0_TXD_P and LANE0_TXD_N	Transceiver transmitter differential outputs
LED_OUT[7:0]	Signal that indicates whether link is up or down
TX	Transmitter of UART interface

3.2. IP Configuration (Ask a Question)

The hardware design for the JESD204B interface includes the following blocks.

3.2.1. Data Handle (Ask a Question)

The data handle (DATA_HANDLE_0) block receives the input data selection and link or data error generation information from the GUI. This block also sends the data output received from the CoreJESD204BRX core and the data or link status error to the GUI for viewing.

3.2.2. Data Generator (Ask a Question)

The data generator has a PRBS generator and a waveform generator. The PRBS generator generates PRBS7, PRBS15, PRBS23 and PRBS31 patterns. An error insertion mode implemented in the PRBS generator inserts an error into the PRBS sequence. The waveform generator generates sine, sawtooth, triangle and square waveforms. The data generator feeds a 64-bit test pattern to the JESD204BTX core, which subsequently transmits the data to the transceiver.

3.2.3. PF_TPSRAM (Ask a Question)

There are two instances of PF_TPSRAM blocks, the PF_TPSRAM_C0 block stores the JESD204B link status before sending it to the GUI. The PF_TPSRAM_C1 block stores the data received from the CoreJESD204BRX before sending the data to the GUI.

3.2.4. Error Generator (Ask a Question)

The error generator block (ERR_GEN_0) generates link errors by sending random data between CoreJESD204BTX and PF_XCVR when link error generation is selected in the GUI.

3.2.5. PRBS_checker (Ask a Question)

The data checker receives 64-bit data from the CoreJESD204BRX IP core and checks whether the received data is correct. It generates an error count and a status signal, which are transmitted to the GUI for status indication. The data checker exclusively checks the PRBS sequences generated by the data generator.

3.2.6. LED Debug (Ask a Question)

The LED debug block (LED_DEBUG_BLK_0) debugs the JESD204B link status and other errors. When the link is up, LEDs 1, 2, 3, 4, 5 and 6 glow, while LEDs 7 and 8 do not glow (with DIP 1, 2, 3 and 4 are set to low on the SW8 dip slide switch).

3.2.7. Init_monitor (Ask a Question)

When the DEVICE_INIT_DONE signal from Init_monitor block goes high, the transceiver is completely configured. This signal is anded with ARST_N signal to get proper reset signal for the design.

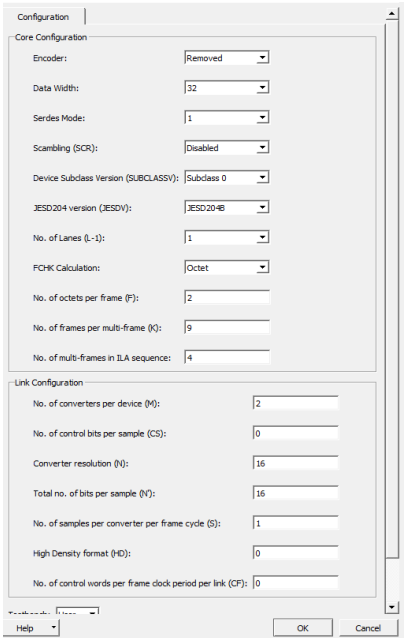
3.2.8. CORERESET_PF (Ask a Question)

CoreReset_PF synchronizes resets to the user-specified clock domain. This ensures that while the assertion is asynchronous, the negation is synchronous with the clock.

3.2.9. CoreJESD204BTX (Ask a Question)

CoreJESD204BTX is the transmitter interface of the JEDEC JESD204B standard. For this demo design, this IP core is configured in Libero®, as shown in the following figure.

Figure 3-3. CoreJESD204BTX Configurator



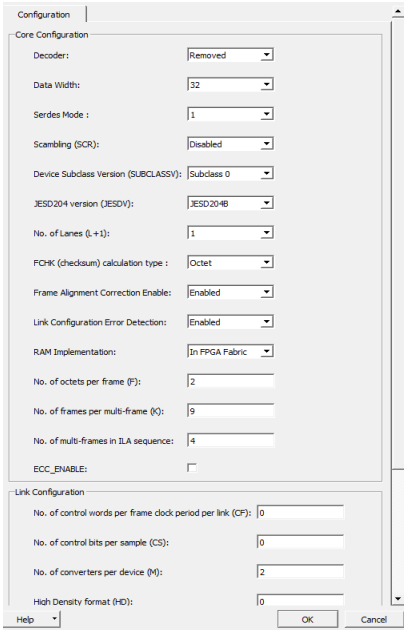
For more information about CoreJESD204BTX, see CoreJESD204BTX Handbook.

3.2.10. CoreJESD204BRX (Ask a Question)

CoreJESD204BRX is the receiver interface of the JEDEC JESD204B standard. For this demo design, this IP core is configured in Libero®, as shown in the following figure.

Note: To view the complete configuration, open the configurator of IP from within the design.

Figure 3-4. CoreJESD204BRX Configurator



For more information about CoreJESD204BRX, see CoreJESD204BRX Handbook.

3.2.11. Transceiver Interface (Ask a Question)

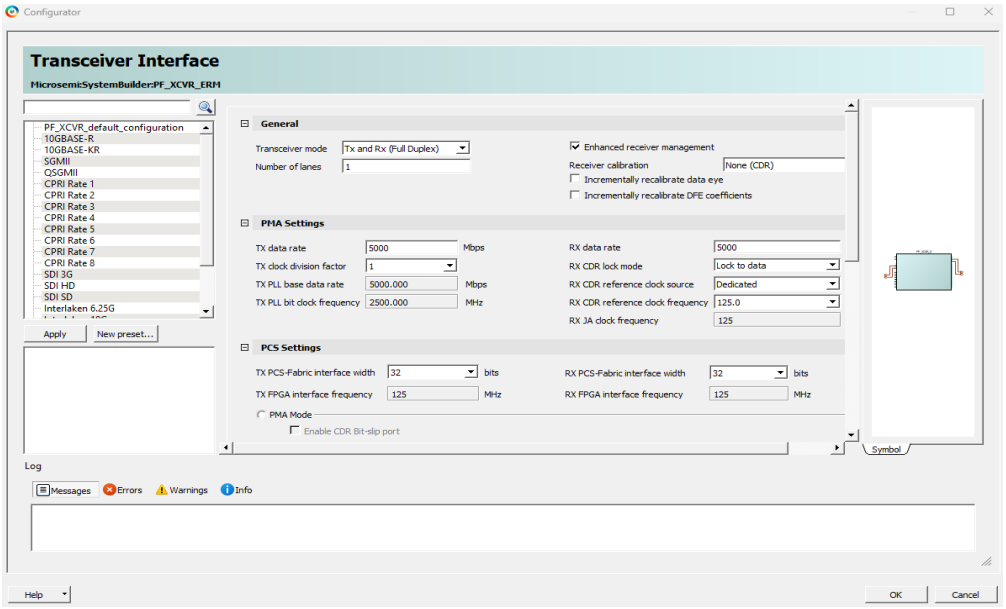
The PolarFire® high-speed transceiver (PF_XCVR) is a hard IP block designed to support high-speed data rates ranging from 250 Mbps to 12.5 Gbps. In this demo, the transceiver block (PF_XCVR) is configured in 8b10b mode with a Clock Data Recovery (CDR) reference clock of 125 MHz to support 5.0 Gbps data rate.

The PolarFire transmit PLL (PF_TX_PLL) provides 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 5.0 Gbps data rate.

The following figure shows the transceiver interface configuration.

Note: To view the complete configuration, open the configurator of IP from within the design.

Figure 3-5. Transceiver Interface Configurator



4. Clocking Structure (Ask a Question)

In the reference design, there are three clock domains:

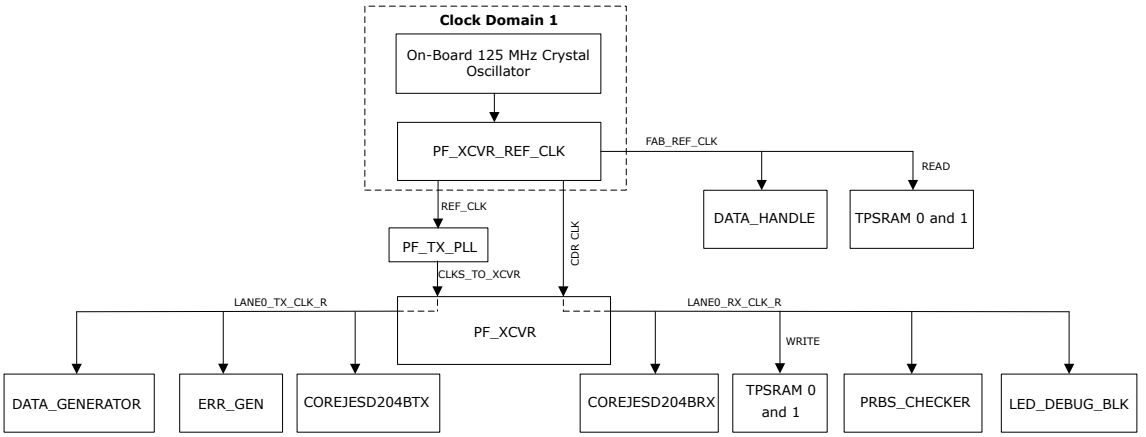
- RX_CLK (125 MHz)
- TX_CLK (125 MHz)
- FAB_REF_CLK (125 MHz)

The on-board 125-MHz crystal oscillator drives the XCVR reference clock, which provides clock to the DATA_GENERATOR, CoreJESD204BTX, ERR_GEN, CoreJESD204BRX, LED_DEBUG, PRBS_CHECKER, TPSRAM C0 & C1 and DATA_HANDLE.

Important: If there is a change in the data rate or reference clock of the transceiver, you must reconfigure COREUART.

The following figure shows the clocking structure.

Figure 4-1. Clocking Structure



5. Reset Structure (Ask a Question)

The DEVICE_INIT_DONE and external reset signal ARST_N are mapped to pin N4 on the Splash Kit. These signals initiate the system reset (FABRIC_RESET_N) through the res_syn_0 block.

The FABRIC_RESET_N signal from the res_syn_0 block provides a direct reset to the following modules:

- CoreJESD204BRX
- CoreJESD204BTX
- PF_XCVR (LANE0_PMA_ARST_N)

Additionally, FABRIC_RESET_N is connected to the reset synchronizer block, which distributes synchronized reset signals to the following functional blocks:

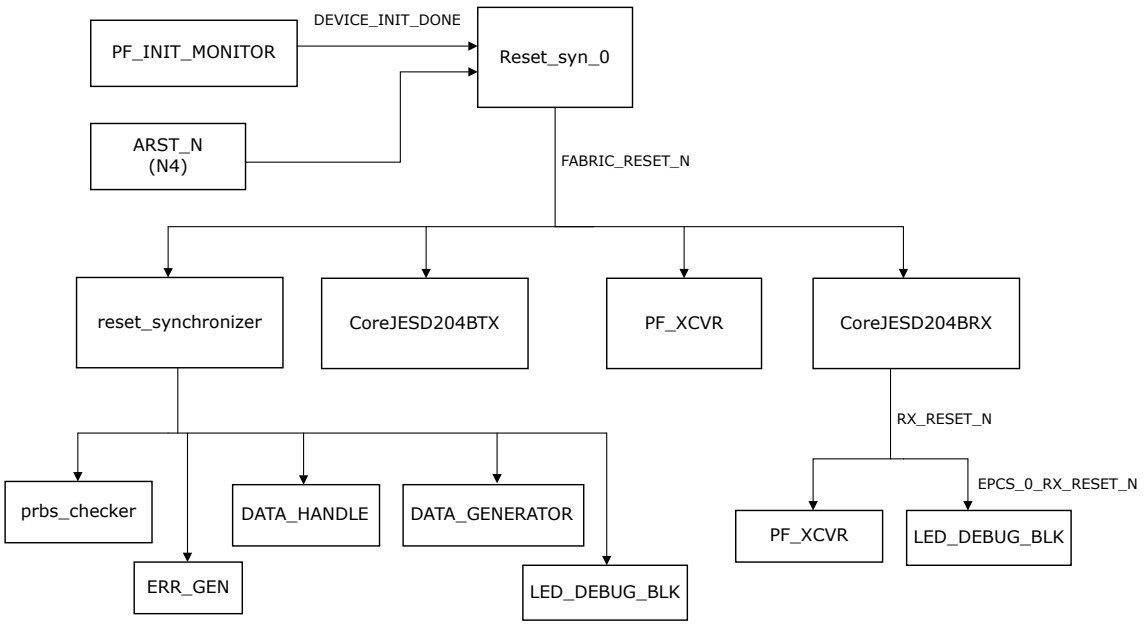
- prbs_checker
- DATA_HANDLE
- DATA_GENERATOR
- ERR_GEN
- LED_DEBUG_BLK

RX_RESET_N output from the CoreJESD204BRX module supplies reset signals to:

- LANE0_PCS_ARST_N input of the PF_XCVR_0 module
- LED_DEBUG block (EPCS_0_RX_RESET_N)

The following figure shows the reset structure.

Figure 5-1. Reset Structure



6. Simulating the PolarFire® JESD204B Design (Ask a Question)

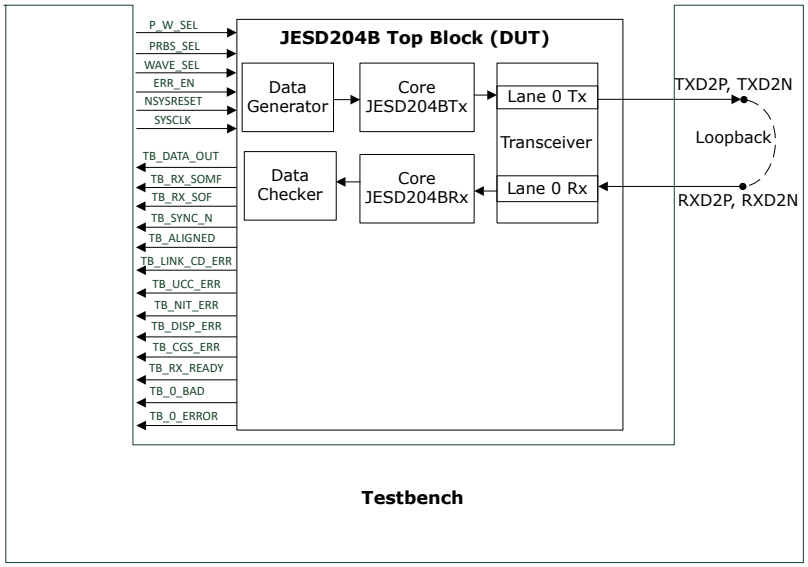
To simulate the design, perform the following steps:

1. Start Libero®, and select **Project** > **Tool Profiles...**.
2. In the **Tool Profiles** window, select **Synthesis** and **Simulation** on the **Tools** panes and select the latest active installation directory paths for these two tools.

For Simulation, browse the design files folder, create Libero Project using provided TCL scripts, and click **Simulate** as highlighted in the Figure 6-2. For more information, see Appendix B: Running the TCL Script.

A testbench is provided to simulate the JESD204B PRBS pattern and waveform selection. The following figure shows the interaction between testbench and the design.

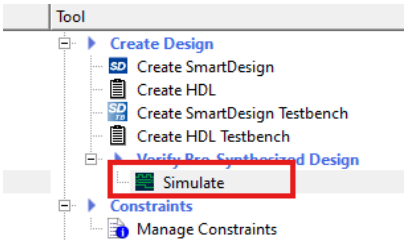
Figure 6-1. Testbench and JESD204B Demo Design Interaction



The testbench generates the test selection for the PRBS input (PRBS7, PRBS15, PRBS23 and PRBS31) and waveform input (sine wave, sawtooth wave, triangle wave and square wave). It also monitors the JESD204B output status signals (SYNC_N, ALIGNED and CGS_ERR) for the verification of JESD204B phases, and PRBS checker output status signals O_BAD and O_ERROR[4:0].

To simulate the design, in the **Design Flow** tab, double-click **Simulate** under **Verify Pre-Synthesized Design**. The **Simulate** option is highlighted in the following figure.

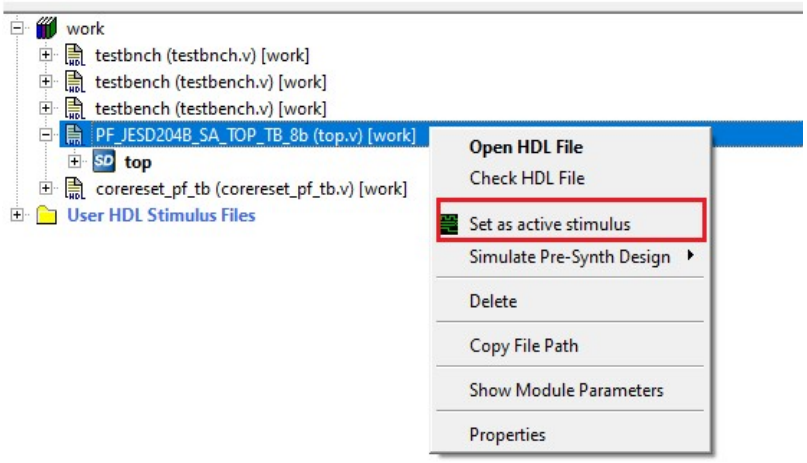
Figure 6-2. Simulating the Design



When the simulation is initiated, simulation tool compiles all the design source files, runs the simulation, and configures the waveform viewer to show the simulation signals.

Note: In certain cases, a prompt may appear asking for the selection of an active stimulus before starting the simulation. To resolve this, navigate to the **Stimulus Hierarchy**, right-click **PF_JESD204B_SA_TOP_TB_8b (top.v)** and select **Set as Active Stimulus**, as shown in the following figure.

Figure 6-3. Set As Active Stimulus



6.1. Simulation Flow (Ask a Question)

The following steps describe the JESD204B testbench simulation flow:

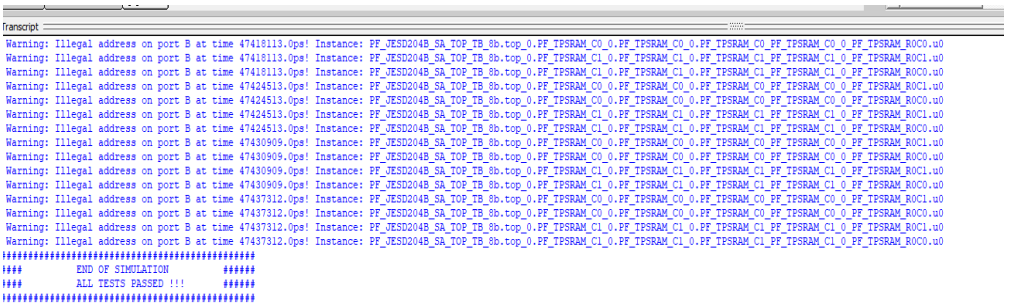
1. At the start, the NSYSRESET signal resets all of the components.
2. After the transceiver block is initialized, the TB_RX_READY signal is asserted high.
3. The JESD204BRX issues a synchronization request by driving the TB_SYNC_N pin low.
4. The JESD204BRX block checks the k28.5 characters transmitted by the JESD204BTX block.
5. The CGS and ILA phase starts after the TB_SYNC_N signal is asserted high.
6. The testbench checks whether the CGS_ERR signal asserts low or not, and completes the code group synchronization phase.
7. The JESD204BRX link asserts the TB_SYNC_N signal to high.
8. After the successful completion of the CGS phase, the JESD204BTX block starts the Initial Lane Alignment (ILA) sequence by transmitting four multi-frames in the following sequence:
   - First frame at TB_TX_SOMF = 0x8
   - Second frame at TB_TX_SOMF = 0x2
   - Third frame at TB_TX_SOMF = 0x8

  - Fourth frame at TB_TX_SOMF = 0x2

9. The JESD204BRX link starts receiving four multi-frames in the following sequence:
   - First frame at TB_TX_SOMF = 0x8
   - Second frame at TB_TX_SOMF = 0x2
   - Third frame at TB_TX_SOMF = 0x8
   - Fourth frame at TB_TX_SOMF = 0x2
10. The ILA phase test passes if all JESD204BRX DATA_OUT is properly received with frame alignment.
11. After successful completion of the ILA phase, the JESD204BTX block enters into the data phase.
12. In the data phase, the following data is fed to the JESD204BTX block:
    PRBS7, PRBS15, PRBS23 and PRBS31 using the PRBS generator.
13. Sine, Square, Saw and triangular waves are generated from the waveform generator.
14. The PRBS checker checks the received PRBS pattern against the expected PRBS pattern.
15. The waveform output can be viewed in the simulation window on corresponding wave selection as shown in Figure 6-5.
16. If the data checker does not detect any error, the testbench issues a TESTBENCH PASSED message stating that the simulation was successful. If an error is detected, the testbench issues a TESTBENCH FAILED message to indicate that the testbench has failed.

While the simulation is running, you can see the status of the test cases in the **Transcript** window of ModelSim, as shown in the following figure.

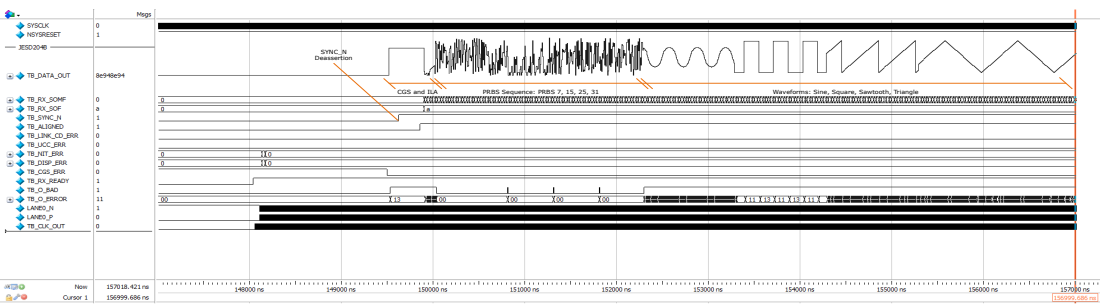
Figure 6-4. Transcript Window



After simulation, the **Waveform** window displays the simulation waveforms as shown in the following figure.

Note: You may notice some warnings in the log. These appear because UART is not used in the simulation. The simulation is focused only on JESD, while UART and RAM are included for GUI purposes.

Figure 6-5. Simulation Waveform Window



7. Setting Up the Demo (Ask a Question)

After generating the bitstream, the PolarFire® device must be programmed. To program the PolarFire device, perform the following steps:

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

Table 7-1. Jumper Settings

Jumper	Description	Default
J11	Close pin 1 and 2 for programming through the FTDI chip. Open pin 1 and 2 for programming through an external FlashPro4 or FlashPro5 device.	Closed
J3	Jumper to select the core voltage. Close pin 1 and 2 for 1.05 V. Open pin 1 and 2 for 1.0 V.	Open
J10	Close pin 1 and 2 for programming through the external SPI flash. If J10 is open, it allows SPI slave programming using the FTDI chip.	Open

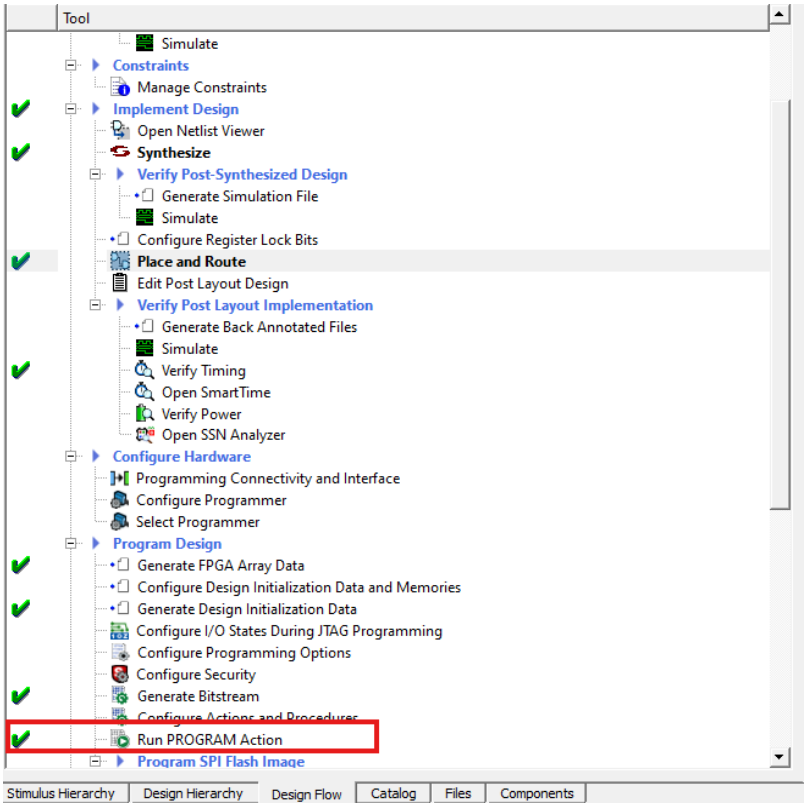
2. Connect the power supply cable to the J2 connector on the board.
3. Connect the USB cable from the host PC to the J1 (FTDI port) on the board.
4. Power On the board using the SW1 slide switch.

   When the board is powered up, power supply LEDs 1 to 4 glow. For more information about LEDs on the PolarFire Splash Board, see UG0786: PolarFire FPGA Splash Kit User Guide.
5. In **Libero Design Flow** tab, double-click **Run PROGRAM Action**.

To view the corresponding log file, navigate to **Reports** tab, right-click **Run Program Action** and select **View Report**.

When the device is successfully programmed, a green tick mark appears as shown in the following figure. For information about how to run the JESD204B standalone demo, see Running the Demo.

Figure 7-1. Device Programming Completed



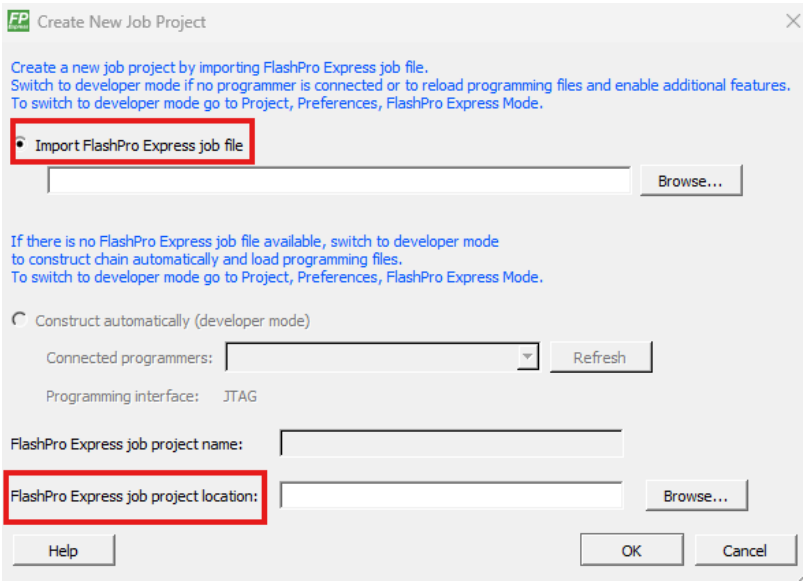
8. Programming the Device Using FlashPro Express (Ask a Question)

This section describes how to program the PolarFire® device with the programming job file using FlashPro Express. The `.job` file is available at the following design files folder location: `mpf_an5978_df/Programming_Files/top.job`.

To program the device, perform the following steps:

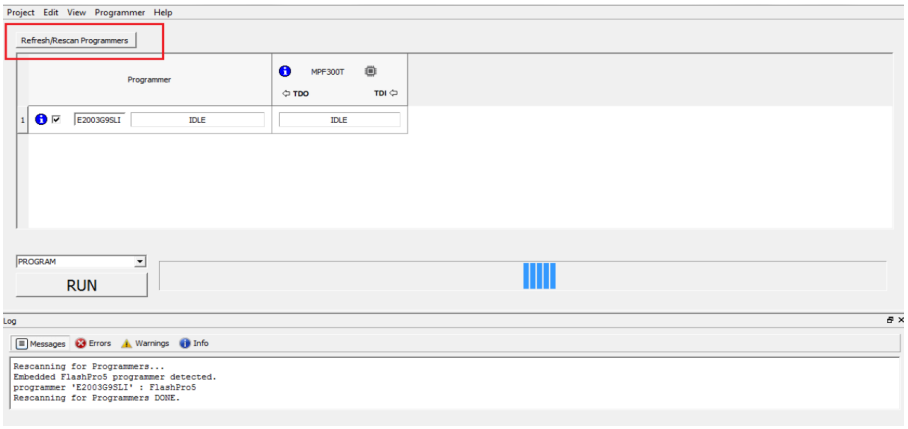
1. On the host PC, launch the FlashPro Express software.
2. To create a new project, click **New** or **New Job Project from FlashPro Express Job** from **Project** menu.
3. Enter the following in the **New Job Project from FlashPro Express Job** dialog box:
   - Programming job file: Click **Browse** and navigate to the location where the job file is located and select the file. The default location is: `mpf_an5978_df/Programming_Files/top.job`.
   - FlashPro Express job project location: Click **Browse** and navigate to the FlashPro Express project location.

Figure 8-1. New Job Project from FlashPro Express Job



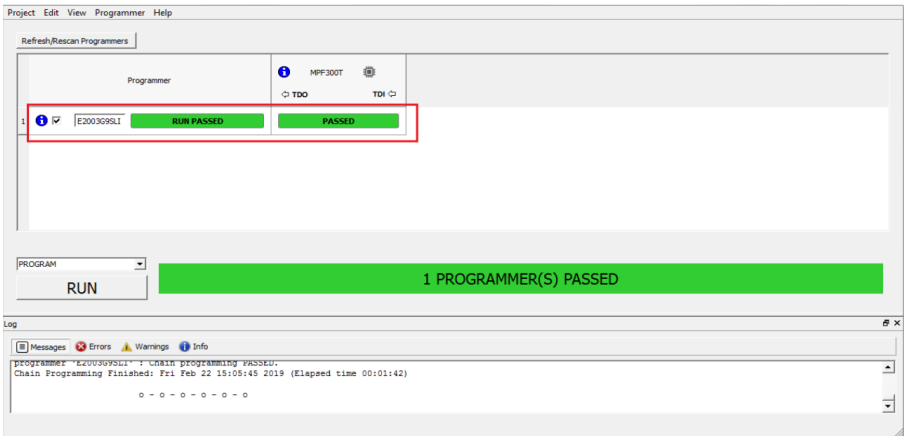
4. Click **OK**. The required programming file is selected and ready to be programmed in the device.
5. The FlashPro Express window appears, as shown in the following figure. Confirm that a programmer number appears in the **Programmer** field. If not, confirm the board connections and click **Refresh/Rescan Programmers**.

Figure 8-2. Programming the Device



6. Click **RUN**. When the device is programmed successfully, a **RUN PASSED** status is displayed as shown in the following figure.

Figure 8-3. FlashPro Express—RUN PASSED



7. Close **FlashPro Express** or click **Exit** in the **Project** tab.

9. Running the Demo (Ask a Question)

This section describes how to use the JESD204B GUI to run the JESD204B demo on the PolarFire® Splash Board.

9.1. Installing the GUI (Ask a Question)

To run the demo, install the JESD204B GUI. The GUI allows selection of different PRBS test patterns as input, and displays the JESD204B status signals and the PRBS status received from the board. The Waveform tab of the GUI displays the output waveforms received from the board for each waveform selected as input.

To install the GUI, perform the following steps:

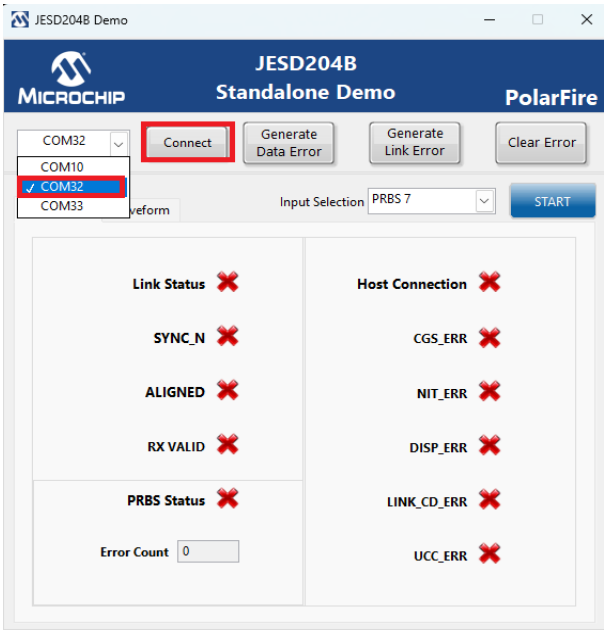
1. Install the JESD204B_GUI application (`setup.exe`) from the following design files folder: `mpf_an5978_df/GUI`.
2. To start the GUI application, double-click the JESD204B_GUI application from the installation directory.

9.2. Running the Demo Design (Ask a Question)

To run the JESD204B demo, perform the following steps:

1. Connect the jumpers and set up the PolarFire® Splash Board as described in steps 1 to 4 of Setting Up the Demo.
2. In **Device Manager** on the host PC, note the COM port associated with the USB serial converter C. To determine the COM port, check the **Location** field in the properties of each COM port.
3. On the **Start** menu of the host PC, click **JESD204B_GUI**.
4. From the list of COM ports, select the COM port identified in the step 2, and click **Connect**, as shown in the following figure.

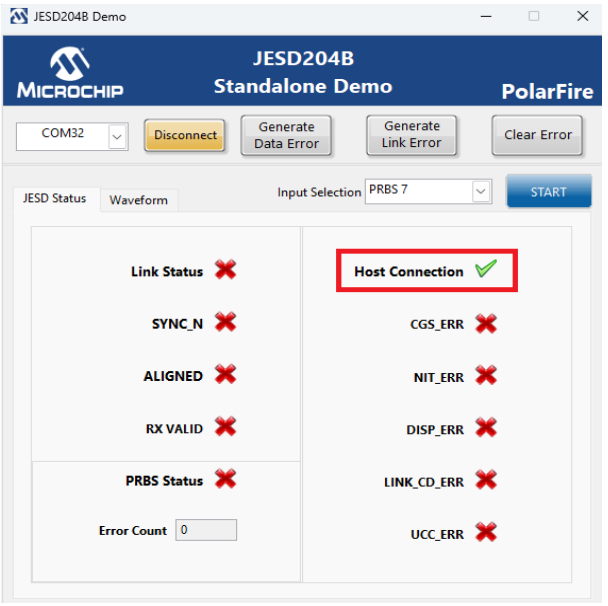
Figure 9-1. COM Port Selection



Important: Port numbers may vary. In this example, COM port 32 is the correct port to select.

After successful connection, the **Host Connection** indicator turns green, as shown in the following figure.

Figure 9-2. Successful Host Connection



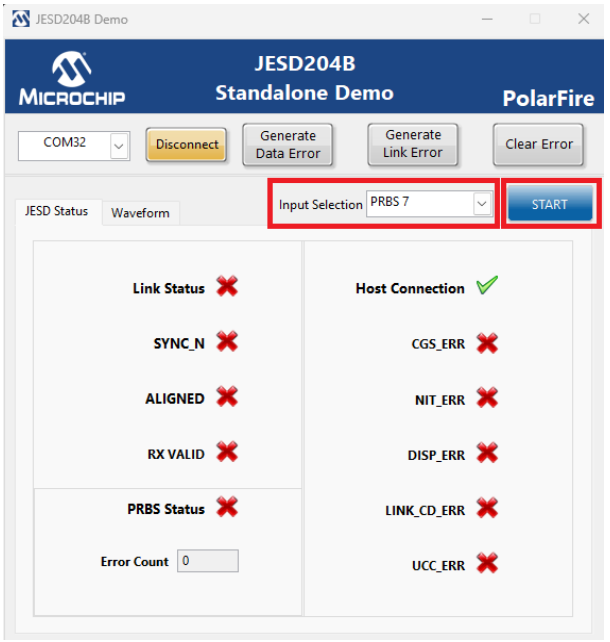
The following table lists the status signals displayed in the JESD204B GUI.

Table 9-1. Status Signals in JESD204B GUI

Signal	Description
Host Connection	Shows the UART communication status.
Link Status	Shows the communication link status between TX and RX.
SYNC_N	Indicates the JESD204B status.
ALIGNED	Indicates that all transceiver lanes are aligned.
RX VALID	Indicates that RX data is valid. In 8b10b mode, indicates that comma alignment has occurred and the CDR is locked.
PRBS Status	Indicates PRBS error.
Error Count	Provides the number of errors that occurred during PRBS check
CGS_ERR	Indicates a code group synchronization error.
NIT_ERR	Indicates a "not in table" error.
DISP_ERR	Indicates a disparity error.
LINK_CD_ERR	Indicates a link configuration data mismatch.
UCC_ERR	Indicates an "unexpected control character" error.

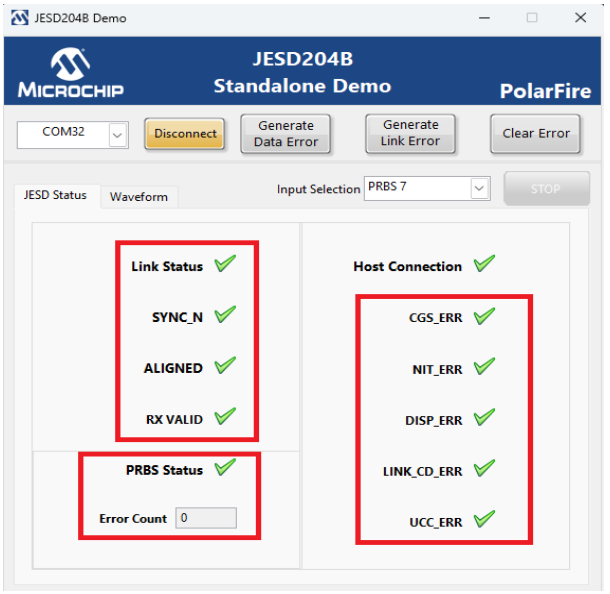
5. From the **Input Selection** list, select the pattern to be transmitted, and click **START**, as shown in the following figure.

Figure 9-3. Pattern Selection



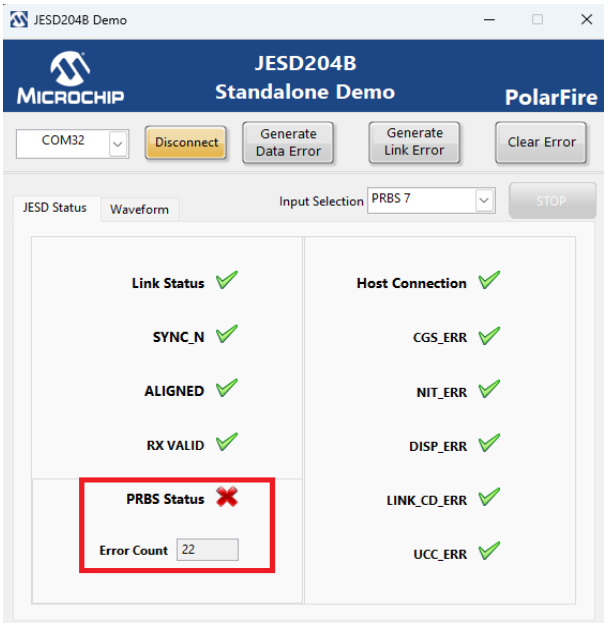
The selected pattern is sent over the serial transmit link and received by CoreJESD204BRX, which checks for errors. At any time, the JESD204B status can be monitored using the status signals on the GUI, as shown in the following figure.

Figure 9-4. Link Status and JESD204B Status



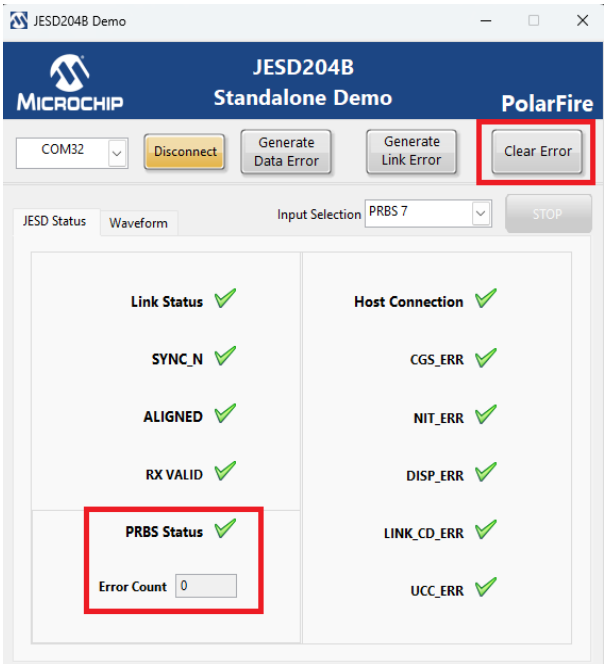
6. To generate an error in the PRBS data, click **Generate Data Error**.
   The **PRBS Status** indicator turns red, and the **Error Count** field displays the number of errors, as shown in the following figure.

Figure 9-5. Data Error



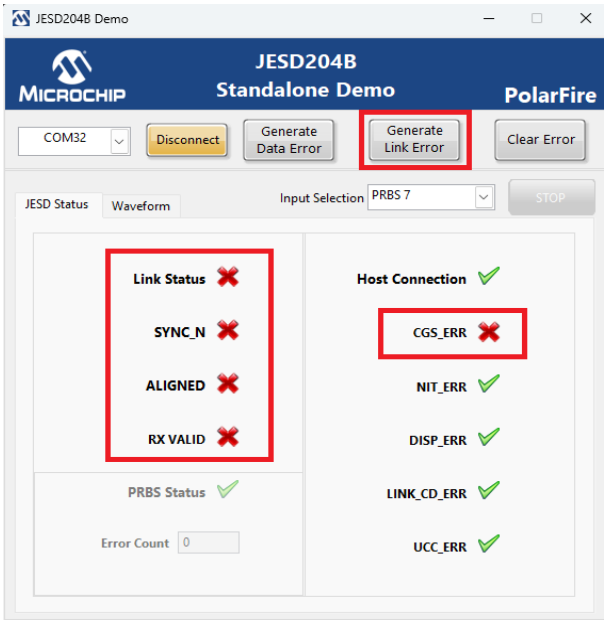
7. Click **Clear Error** to clear the errors in the PRBS data and reset the PRBS status.
   The **PRBS Status** indicator turns green, and the **Error Count** changes to 0, as shown in the following figure.

Figure 9-6. Data Error Cleared



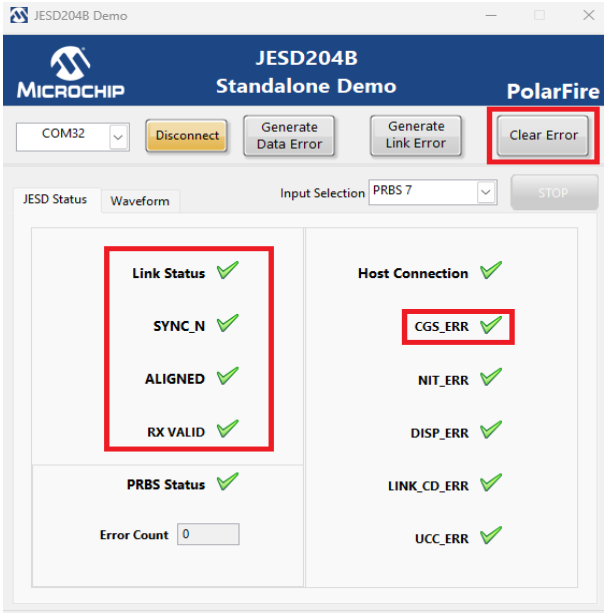
8. To generate a link error between CoreJESD204BTX and the transceiver lane, click **Generate Link Error**.
   The Link Status, SYNC_N, ALIGNED, RX VALID, DISP_ERR and CGS_ERROR indicators turn red, as shown in the following figure.

Figure 9-7. Link Error



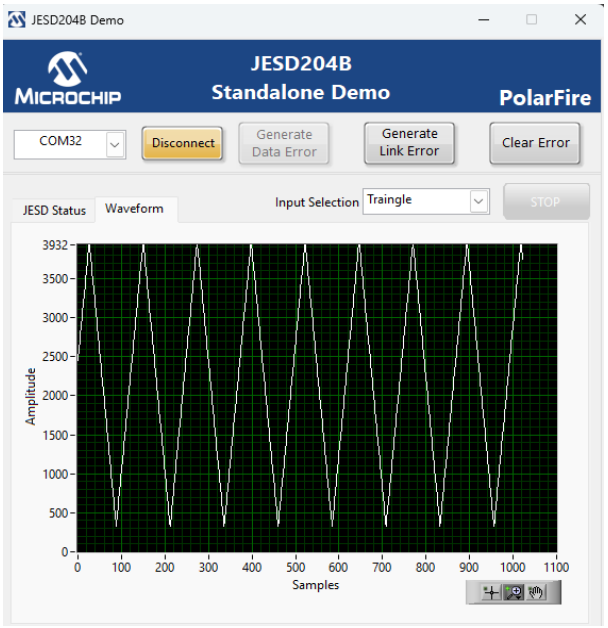
9. To clear the link error, click **Clear Error**.
   The status indicators turn green, as shown in the following figure.

Figure 9-8. Clear Link Error



10. To change the pattern, select **Triangle** from the **Input Selection** list.
    The selected pattern is sent over the serial transmit link and received by CoreJESD204BRX. At any time, the JESD204B status can be monitored using the status signals on the GUI.
11. To view the waveform received from CoreJESD204BRX, click the **Waveform** tab, as shown in the following figure.

Figure 9-9. Triangle Waveform



12. To end the demo, click **Stop** and close the GUI.

10. Appendix A: References (Ask a Question)

This section lists documents that provide more information about the JESD204B standard and IP cores used in the demo design.

- For information about the JESD204B interface standard, visit the JEDEC website.
- For information about PolarFire transceiver blocks, PF_TX_PLL and PF_XCVR_REF_CLK, see PolarFire Family Transceiver User Guide.
- For more information about PF_TPSRAM (PF Micro SRAM), see PolarFire Family Fabric User Guide.
- For more information about CoreJESD204BTX, see CoreJESD204BTX Handbook.
- For more information about CoreJESD204BRX, see CoreJESD204BRX Handbook.
- For more information about Libero, ModelSim and Synplify, see the Microchip Libero SoC webpage.

11. Appendix B: Running the TCL Script (Ask a Question)

TCL scripts are provided in the design files folder under directory `HW`. If required, the design flow can be reproduced from Design Implementation till generation of job file. To run the TCL, perform the following steps:

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

After successful execution of TCL script, Libero project is created within `HW` directory. For more information about TCL scripts, see `mpf_an5978_df/HW/TCL_Script_readme.txt`.

For more details on TCL commands, see TCL Commands Reference Guide. For any queries encountered when running the TCL script, contact Technical Support.

12. Revision History (Ask a Question)

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

Revision	Date	Description
A	08/2025	The following is the list of changes made in the revision A of the document: • The document was migrated to the Microchip template. • The document number was updated from 50200796 to DS00005978. • The document ID was updated from DG0796 to AN5978.
3.0	—	This document is updated with respect to Libero® SoC PolarFire v2.2 release.
2.0	—	This document is updated with respect to Libero SoC PolarFire v2.1 release.
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. 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

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-1709-6

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.

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.

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 products are 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.