DG0757 Demo Guide PolarFire FPGA 10GBASE-R Ethernet Loopback





а 🔨 Міскоснір 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 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	Revisi	on History	1
	1.1	Revision 8.0	1
	1.2	Revision 7.0	1
	1.3	Revision 6.0	1
	1.4	Revision 5.0	1
	1.5	Revision 4.0	1
	1.0	Revision 2.0	י 2
	1.8	Revision 2.0	2
2	PolarF	ire FPGA 10GBASE-R Ethernet Loopback	3
	2.1	Design Requirements	3
	2.2	Prerequisites	4
	2.3	Demo Design	4 5
		2.3.2 Design Blocks and IP Configuration	6
		2.3.3 Clocking Structure	1
		2.3.4 Reset Structure	1
	2.4	Simulating the 10GBASE-R Ethernet Loopback Design	2
		2.4.2 Design Description	3
		2.4.3 Design Implementation	4
		2.4.4 Simulation Flow	5
3	Libero	Design Flow	9
	3.1	Synthesize	9
	3.2	Place and Route	0
	2.2	3.2.1 Resource Utilization	1
	3.3 3.4	Generate EPGA Array Data	י 1
	3.5	Configure Design Initialization Data and Memories	2
	3.6	Generate Bitstream	2
	3.7	Run Program Action	3
4	Runnii	ng the Demo	4
5	Appen	dix 1: Programming the Device Using FlashPro Express	7
6	Appen	dix 2: Enabling SyncE in 10G BaseR Design	C
	6.1	Prerequisite	0
	6.2	Design Implementation	1
		6.2.1 Transceiver	2
	0.3 6.4	Libero Design Flow	3 2
	6.5	Running the Demo	4
	0.0		Ŧ
7	Appen	idix 3: References	3



Figures

Figure 1	Hardware Implementation Block Diagram	5
Figure 2	10GBASE-R Loopback Hardware Design Libero Implementation	5
Figure 3	CORE10GMAC Configuration	7
Figure 4	Transceiver Interface Configuration	8
Figure 5	Transmit PLL Configuration	9
Figure 6	Transceiver Reference Clock Configuration	9
Figure 7	Clocking Structure	1
Figure 8	Reset Structure	2
Figure 9	Testbench and 10GBASE-R Ethernet Loopback Design Interaction	3
Figure 10	Libero Implementation of the top SmartDesign Module	4
Figure 11	Libero Implementation of the top-tb SmartDesign Module 1	5
Figure 12	Use Automatic DO File Option Selected 1	5
Figure 13	O_CORE_RX_SRESET and O_CORE_RX_SRESET at 0 1	6
Figure 14	Ethernet Packet Sent	7
Figure 15	Good Packets Count Incremented by 1 1	8
Figure 16	Libero Design Flow Options 1	9
Figure 17	Edit with I/O Editor Option	20
Figure 18	I/O Editor Transceiver View	20
Figure 19	Component Locations Updated in user_fp.pdc File 2	21
Figure 20	Design and Memory Initialization Window 2	22
Figure 21	PolarFire Evaluation Board Setup	23
Figure 22	10G Ethernet Settings	24
Figure 23	Link Status	24
Figure 24	Stream Block Addition	25
Figure 25	Stream Block Added	25
Figure 26	Port Load Settings	26
Figure 27	Ethernet Traffic Data	26
Figure 28	FlashPro Express Job Project 2	27
Figure 29	New Job Project from FlashPro Express Job	28
Figure 30	Programming the Device	28
Figure 31	FlashPro Express—RUN PASSED	29
Figure 32	Hardware Implementation	30
Figure 33	10GBASE-R Loopback Hardware Design Libero Implementation —SyncE 3	31
Figure 34	Transceiver—SyncE	32
Figure 35	Transmit PLL—SyncE	33
Figure 36	Frequency Offset—SyncE Disabled	34
Figure 37	Frequency Offset Between TX and RX 3	35
Figure 38	Frequency Offset—SyncE Enabled	35



Tables

Table 1	Design Requirements	:	3
Table 2	Hardware Design Clock Frequencies	4	1
Table 3	Core10GMAC Configuration	8	3
Table 4	Simulation Signals	14	4
Table 5	Resource Utilization	2′	1
Table 6	Jumper Settings for PolarFire Device Programming	23	3



1 Revision History

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

1.1 Revision 8.0

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

- Updated the document for Libero SoC v12.2.
- Removed the references to Libero version numbers.

1.2 Revision 7.0

Updated the document to include features and enhancements introduced in the Libero SoC v12.0.

1.3 Revision 6.0

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

- Updated the document to include features and enhancements introduced in the Libero SoC PolarFire v2.2 release.
- Added new Appendix for Enabling SyncE in 10G BaseR Design. For more information, see Appendix 2: Enabling SyncE in 10G BaseR Design, page 30.

1.4 Revision 5.0

Updated the document for Libero SoC PolarFire v2.1 release.

1.5 Revision 4.0

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

- The document was updated to include features and enhancements introduced in the Libero SoC PolarFire v2.0 release.
- The design requirements were updated. For more information, see Design Requirements, page 3.
- Details about the demo design, including the hardware implementation block diagram, were updated. For more information, see Demo Design, page 4.
- Libero design implementation details were updated. For more information, see Figure 2, page 5.
- IP configuration details were updated. For more information, see 10GBASE-R Loopback Hardware Design Libero Implementation, page 5.
- Clocking Structure diagram was added. For more information, see Figure 7, page 11.
- A new section which details the reset structure of the design is added. For more information, see Reset Structure, page 11.
- Reset Structure diagram is added. For more information, see Figure 8, page 12.
- Information about simulating the design was updated. For more information, see Simulating the 10GBASE-R Ethernet Loopback Design, page 12.

1.6 Revision 3.0

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

- The document was updated to include features and enhancements introduced in the Libero SoC PolarFire v1.1 SP1 release.
- Hardware requirements were added, and software requirements were updated to include Spirent TestCenter and FlashPro. For more information, see Design Requirements, page 3.
- Information about how to program the device was added. For more information, see Programming the Device Using FlashPro, page 25.
- Information about how to run the hardware reference design was added. For more information, see Running the Demo, page 24.



1.7 Revision 2.0

•

•

The following is a summary of the changes made in revision 2.0 of this document.

- The document was updated for Libero SoC PolarFire v1.1 release.
- Information about resource utilization was added. For more information, see Resource Utilization, page 21.

1.8 Revision 1.0

The first publication of this document.



2 PolarFire FPGA 10GBASE-R Ethernet Loopback

The PolarFire[®] FPGA 10G Ethernet solution is compliant to IEEE 802.3ae standard, which supports data transfer rates up to 10.3125 Gbps. The advantages offered using PolarFire FPGAs for building 10G Ethernet solutions include: the use of low-power transceivers, low-power FPGA fabric, and an in-built SyncE-compliant jitter attenuation.

The 10G Ethernet solution is implemented using the CORE10GMAC soft IP Media Access Control (MAC) core, which can be configured either in 10GBASE-KR mode or 10GBASE-R mode.

This demo design includes the following two designs, which can be used as reference designs for building a 10GBASE-R Ethernet loopback application:

- a 10GBASE-R Ethernet 64-bit loopback design that can be used for simulation
- a 10GBASE-R Ethernet 32-bit loopback design that can be run on the PolarFire Evaluation Board using Spirent TestCenter

These demo designs can be programmed using either of the following options:

- Using the .job file: To program the device using the .job file provided with the design files, see Appendix 1: Programming the Device Using FlashPro Express, page 27
- Using Libero SoC: To program the device using Libero SoC, see Libero Design Flow, page 19. Use this option when the demo design is modified.

2.1 Design Requirements

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

Table 1 •	Design Requirements
-----------	---------------------

Requirement	Version			
Operating system	Windows 7 or Windows 10			
Hardware				
PolarFire Evaluation Kit (MPF300-EVAL-KIT)	Rev D or later			
Spirent test module for 10G Ethernet traffic generation				
Optical fiber cable				
Small form-factor pluggable (SFP+) module				
Software				
Libero SoC				
Synplify Pro	Note: Refer to design t	the readme.txt file provided in the files for the software versions		
Spirent TestCenter	used wi	th this reference design.		
FlashPro Express				
ModelSim				

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



2.2 **Prerequisites**

Before you begin:

- For demo design files download link: http://soc.microsemi.com/download/rsc/?f=mpf_dg0757_df
- 2. Download and install Libero SoC (as indicated in the website for this design) on the host PC from the following location:

https://www.microsemi.com/product-directory/design-resources/1750-libero-soc#downloads

- 3. From the Project menu, click **Open** Project.
- 4. Navigate to the mpf_dg0757_df/mpf_10GBaseR_df/Libero_Project folder, select the Libero_Project.prjx file, and click Open.
- 5. Download all the IP cores from Libero SoC > Catalog:

Note: To simulate this demo design, see Simulating the 10GBASE-R Ethernet Loopback Design, page 12.

2.3 Demo Design

The 10GBASE-R Ethernet loopback hardware design loops back the Ethernet traffic generated by the Spirent test module through the CORE10GMAC IP. A FIFO logic is implemented in the RTL to loop the Rx signals of the Core10G MAC back to the Tx signals of the MAC.

This looped back data is sent through the TX interface of the transceiver that is received by the Spirent TestCenter. Using the Spirent TestCenter software, the received data is analyzed for throughput rate and errors in the incoming packets.

The 10GBASE-R Ethernet loopback design includes the following components:

- CORE10GMAC: Serves as a 10-Gbps Ethernet MAC that transmits and receives the Ethernet packets.
- Transceiver: Acts as a 10GBASE-R physical interface for data transfers; configured for 64b/66b encoding/decoding with scrambler/descrambler enabled with a PCS interface width of 32 bits to the CORE10GMAC.
- CoreABC: Configures the CORE10GMAC registers.
- FIFO interface logic: Loops back the CORE10GMAC Rx data to Tx data.
- PF_TX_PLL: Generates the bit clock required for the transceiver.
- PF_XCVR_REF_CLK: Generates the fabric clock and the reference clock for the transceiver and the TX_PLL.

The following table lists the clock frequencies used in the design.

Frequency (MHz)
156.25
5156.25
156.25
322.26
322.26
50

Table 2 • Hardware Design Clock Frequencies



The following figure shows the top-level block diagram of the PolarFire 10GBASE-R Ethernet loopback hardware implementation.





2.3.1 Design Implementation

The following figure shows the Libero implementation of the 10GBASE-R Ethernet loopback hardware design.







2.3.2 Design Blocks and IP Configuration

The following IPs need to be configured before simulating and implementing the demo design.

- Core10GMAC, page 6
- Transceiver Interface, page 8
- Transmit PLL, page 9
- Transceiver Reference Clock, page 9
- CoreABC, page 10
- CoreAPB3, page 10
- PF_POWER_INIT, page 10
- PF_CCC_0, page 10
- FiFo_wrapper_Top, page 10

2.3.2.1 Core10GMAC

Core10GMAC is configured for 10GBASE-R mode with a core data width of 32 bits. Core data width is the width of the data path connected to the transceiver interface. The system data width, that is, the width of the interface to the user logic, is configured as 64 bits. (In this demo, the FiFo_wrapper_top module provides this interface.)

The Tx and Rx Pause features are disabled, and both the MAC TX FIFO depth and MAC RX FIFO depth are set to 256.



The following figure shows the settings selected in the CORE10GMAC Configurator.

Figure 3 • CORE10GMAC Configuration

osemi:DirectCore:COR	E10GMAC:2.2.101				
Configuration M	IAC Tx Stat Counters	MAC Rx Stat Counters			
Personality			_		
System Data Width: 64	• (Core Data Width: 32 💌]		
10G Type: 10G	BASE-R				
Pause Features					
Tx Port/PFC:	Disabled	Rx Port/PFC:		Disabled	 •
Tx Timer Enable:	Γ	Rx Check Pause	Multi-Cast:	4	
Rx Check Pause Unicast-Cast:	Г				
Tx MAC Features					
MAC TX FIFO Depth:	256 💌	MAC TX Preamble:			
TX IFG Count:	Fixed at 12	MAC TX Local Loopback Enable:			
MAC TX Check LT Field:					
Rx MAC Features					
MAC RX FIFO Depth:	256 💌	MAC RX Preamble:			
MAC RX Local Loopback Enable:		MAC RX Check LT Field:			
Rx Global Flow Control:					
PCS 73 Rx Gearbox					
PCS 73 Rx Gearbox Enable:		M			
APB Timeout					
APB Timeout Enable:	APB Timeout Count:	80	-		



The Core10GMAC IP is configured using the CoreABC soft processor. The Core10GMAC configuration for the demo design is as follows:

Table 3 •	Core10GMAC Configuration	
-----------	--------------------------	--

Register	Address	Offset	Bit	Binary Value
MAC Tx Config Register	(0xA)	0x3	cfg_sys_mac_tx_en	1
		0x4	sys_mac_tx_fcs_ins	1
MAC Rx Config Register	(0xB)	0x0	mac_rx_fcs_remove	1
		0x3	cfg_sys_mac_rx_en	1

For information about the features and registers of Core10GMAC, see Libero SoC > Catalog> Core10GMAC Handbook.

2.3.2.2 Transceiver Interface

The PolarFire high-speed transceiver (PF_XCVR) is a hard IP block and supports data rates from 250 Mbps to 12.5 Gbps. In this demo, PF_XCVR is configured for the data rate of 10312.5 Mbps. It is configured with a CDR reference clock of 156.25 MHz with lock to data selected as the CDR lock mode. The PCS of the transceiver is interfaced with CORE10GMAC. It is configured for the 64/66b mode with scrambler/descrambler enabled. The scrambler, which is self-synchronizing, generates sufficient transitions to aid data and clock recovery at the CDR. The following figure shows the transceiver interface configuration.



Configurato σ Transceiver Interface ilder:PF_XCVR_ERM General Tx and Rx (Full Duplex • Enhanced receive Transceiver mode -Number of lanes None (CDR • Receiver cal PF XCVR 0 10312.5 Lock to de PADs OUT 10312.5 Mbps BX data rate Mbp LANE0_TXD_N lock division fac RX CDR lock r CTRL CLK LANE0_TXD_P Mbps 156.25 • MHz CTRL_ARST_N LANE0_OUT CLKS_FROM_TXPLL_0 LANEO_RX_BYPASS_DATA as E PCS PADs IN LANE0_RX_CLK_R TX PCS-Fabric interface width 32 - bits RX PCS-Fabric interface width 32 • bits ►LANE0_RXD_N LANE0_RX_DATA[31:0] 322.266 A interface frequency 322.26 MHz MHz ►LANE0_RXD_P LANE0_RX_DATA_VAL Enable CDR Bit-si LANE0_RX_HDR[3:0] LANE0_IN LANE0_CDR_REF_CLK_0 LANE0_RX_HDR_VAL 🖸 §b 10b Encoding/Decoding €4b6xb Gear Box -€4b66b ►LANE0_LOS LANE0_RX_IDLE C 646676 LANE0_PCS_ARST_N LANE0_RX_READY Enable Disparity Enable BER monitor state m LANEO_PMA_ARST_N LANEO_RX_SOS Enable Scrambler/ 🔽 Enable 32 bits data width LANE0_TX_DATA[31:0] LANE0_RX_VAL Soft PIPE Interface -LANEO STATUS_HI_BER LANE0_TX_HDR[3:0] ►LANE0_TX_SOS LANE0_STATUS_LOCK LANE0_TX_CLK_G LANEO TX CLK STABLE Use as PLL reference dock PF_XCVR Globa • · 0 JX clock BX dock TX ar Ŧ TX and RX ECS Reset • Enable TX_BYPASS_DATA port Enable RX_READY_CDR and RX_VAL_CDR ports • Symbol 😵 Errors 🔺 Warnings 🕕 Infe OK Cancel Help •



2.3.2.3 Transmit PLL

The PolarFire transmit PLL (PF_TX_PLL) is a hard IP block that provides a bit clock and a reference clock to the transceiver block. The transmit PLL is configured with a reference clock of 156.25 MHz and generates an output clock of 10312.5 Mbps.

The following figure shows the transmit PLL configuration.

Figure 5 • Transmit PLL Configuration

Configurator	_		\times
Transmit PLL Microsemi:SgCore:PF_TX_PLL:2.0.002			
Transmit PLL			
Clock Inputs/Outputs			
Reference Clock Source Dedicated			
Desired Output Bit Clock 10312,500 Mbps 5156.250 MHz			
Bandwidth Low T	_TX_PLL	_0_	
Clock Options			
© Normal Mode			
C 3ltter Cleaning Mode 10G SyncE 328H			
E Features			
Enable Dynamic Reconfiguration Interface (DRI)			
▲ Symbol			
Heb	ок	Cancel	

2.3.2.4 Transceiver Reference Clock

The transceiver reference clock (PF_XCVR_REF_CLK) is a hard IP block that provides a reference clock (REF_CLK) of 156.25 MHz to the transmit PLL and a fabric reference clock (FAB_REF_CLK) which is provided as an input to the Clock Conditioning circuit (CCC) to generate the pclk (for configuration) and I_SYS_CLK of the CORE10GMAC.

The following figure shows the transceiver reference clock configuration.

Figure 6 • Transceiver Reference Clock Configuration

Configurator	- 🗆 X
Transceiver Reference Clock Con	nfigurator
Microsemi:SgCore:PF_XCVR_REF_CLK:1.0.103	-
Configuration	
Reference Clock 0	
Enable reference dock 0 🔽	
Reference Clock 0 Mode Differential	
Enable fabric dock output 🔽	
Reference Clock 1	
Enable reference dock 1	
Reference Clock 1 Mode LVCMOS	
Enable fabric dock output 🔽	- REF CIK PAD P REF CIK-
	PE XCVR REE CLK
Help V	Symbol OK Cancel



2.3.2.5 CoreABC

CoreABC is a configurable, low-gate count controller intended for Advanced Microcontroller Bus Architecture Advanced Peripheral Bus (AMBA APB)-based designs. Because this demo design requires only a few registers to be configured, and no dynamic changes are required in the configuration, the CoreABC processor is used in this design. Depending on the application requirements, RISC-V, Cortex-M1, or any other soft processor may be used for configuring the registers.

2.3.2.6 CoreAPB3

CoreAPB3 is a bus component that provides an AMBA AHB fabric for interconnection between an APB master and up to 16 APB slaves. CoreAPB3 supports a single APB3 master. In this design, CoreAPB3 is used to connect the CoreABC APB master interface to the CORE10GMAC APB slave interface.

2.3.2.7 PF_POWER_INIT

The PF_POWER_INIT block ensures the device is powered up systematically. The process of powering up the device includes three steps:

- 1. Power-on reset
- 2. Programmed device boot
- 3. Design initialization

During design initialization, the transceiver configuration is initialized using the data stored in the non-volatile memory. The output of the PF_POWER_INIT block is ANDed with the resets used in the design to reset the entire logic.

2.3.2.8 PF_CCC_0

The PolarFire Clock Conditioning Circuitry (CCC) block takes an input clock of 156.25 MHz from the FAB_REF_CLK signal (output of PF_XCVR_REF_CLK) and generates a 50 MHz clock at OUT0 and 156.25 MHz clock at OUT1. The OUT0 port of the CCC is used for the configuration and OUT1 is used for the user logic in the design.

2.3.2.9 FiFo_wrapper_Top

FiFo_wrapper_Top is a user-defined RTL module, which uses the CoreFIFO IP to loop the MAC RX PACKET INTERFACE to the MAC TX PACKET INTERFACE.



2.3.3 Clocking Structure

This design has one clock domain. The on-board 156.25 MHz crystal oscillator generates the clocks used in the demo design. The clock divider generates 40 MHz clock for the XCVR_ERM. The following figure shows the clocking structure of the design.

Figure 7 • Clocking Structure



2.3.4 Reset Structure

The reset structure of the design is shown in Figure 8, page 12 The output of the **PF_POWER_INIT** monitor ANDed with **PLL_Lock_0** is used to reset the logic in the design. This output is combined with following signals to reset the modules in the design:

- The output is ANDed with O_SYS_MAC_RX_RDY to reset the COREABC and FiFo_wrapper_top module. CoreABC configures the CORE10GMAC when MAC RX is ready.
- The output is NAND with the transceiver control signal, LANE0_RX_VAL, and is used to reset the RX path of the CORE10GMAC. The MAC RX path is held in reset until the transceiver, LANE0_RX_VAL is driven to '1'.

The **PF_POWER_INIT** monitor output, ANDed with the transceiver control signal, **LANE0_TX_STABLE**, issued to reset the **TX** path of the **CORE10GMAC**.

The MAC TX path is held in reset until the transceiver, LANE0_TX_STABLE is driven to '1'. The LANE0_PCS_ARST_N signal of PF_XCVR is reset using O_CORE_TX_SRESET of the CORE10GMAC.



Figure 8 • Reset Structure



2.4 Simulating the 10GBASE-R Ethernet Loopback Design

The following sections list the prerequisites for simulation of the 10GBASE-R Ethernet loopback design, provide details about the design implementation, and describe the simulation flow.

2.4.1 Prerequisites

Before you begin:

- 1. Start Libero SoC, 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.
- 3. From the Project menu, click **Open Project**. The Open Project dialog box opens.
- 4. Navigate to the mpf_dg0757_df/mpf_10GBaseR_df/Simulation_Project/BaseR_Sim folder, select the BaseR_Sim.prjx file, and click Open. The PolarFire 10G Ethernet Simulation design project opens in Libero.
- 5. Download the following IP cores from Libero SoC Catalog:
 - PF_XCVR
 - PF_TX_PLL
 - PF_XCVR_REF_CLK
 - CORE10GMAC
 - CoreABC

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







2.4.2 Design Description

The 10GBASE-R Ethernet loopback simulation design includes the following components:

- **Testbench_10g:** Top testbench module that generates the clocks required for the device under test (DUT) and the testbench submodule, and interconnects the ports from DUT to the testbench submodule.
- top_tb: Testbench submodule, which consists of the following major blocks:
 - CoreABC: Configures 10G MAC registers.
 - packet_generator_checker: Performs the packet generator function of defining the Ethernet frame to be transmitted to CORE10GMAC (packet generator). Performs the packet checker function of receiving the looped back Ethernet frame from the CORE10GMAC and comparing it with the transmitted frame.
 - CORE10GMAC: 10-Gbps Ethernet MAC configured in Base-R mode which transmits and receives the Ethernet packets.
- top: DUT block of the design, which consists of the following major blocks:
 - CORE10GMAC: 10-Gbps Ethernet MAC configured in Base-R mode which transmits and receives the Ethernet packets.
 - Transceiver: 10GBASE-R physical interface for data transfers; configured in 64b/66b mode with a PCS fabric width of 64 bits.
 - CoreABC: Configures the CORE10GMAC registers.



• FIFO interface logic: Loops back the Ethernet packets; implemented in Verilog RTLThe following table lists the simulation signals transmitted between the Testbench_10g, top, and top tb blocks.

Table 4 • Simulation Signals

Output From	Input To	Signal	Description	
testbench_10g	top, top_tb	SYSCLK	156.25-MHz clock	
		OUT0_FABCLK_0_net_0	50-MHz clock	
top	testbench_10g	LANE0_RXD_P	Rx port signals	
		LANE0_RXD_N	_	
		LANE0_TXD_P	Tx port signals	
		LANE0_TXD_N	-	
top_tb	testbench_10g	LANE0_RXD_P_0	Rx port signals	
		LANE0_RXD_N_0	_	
		LANE0_TXD_P_0	Tx port signals	
		LANE0_TXD_N_0	-	

2.4.3 Design Implementation

The following figure shows the Libero implementation of the top SmartDesign module.







The following figure shows the Libero implementation of the top_tb SmartDesign module.

Figure 11 • Libero Implementation of the top-tb SmartDesign Module



2.4.4 Simulation Flow

The design can be simulated using ModelSim ME 10.7C Pro provided with the Libero SoC installation. The following sections describe the simulation flow for each of these ModelSim versions.

2.4.4.1 Initiating Simulation with ModelSim ME 10.7C Pro

1. In Libero SoC, go to **Project > Project Settings > Simulation options > DO file**, and ensure the **Use automatic DO file** check box is selected, as shown in the following figure.

Figure 12 • Use Automatic DO File Option Selected

Project settings		- 🗆	×	
Device selection Device settings Design flow Analysis operating conditions Simulation options Simulation options Variance Variance Variance Variance Simulation ibinaries Polaririe COREAHBUTE_LIB COREAHBUTE_LIB COREAHBUTE_LIB COREAHBUTE_LIB COREAPB3_LIB	1us Image: Constraint of the second	Restore De	faults	



2.4.4.2 Simulation Results

When the simulation is initiated, ModelSim compiles all the design source files, runs the simulation, and launches the waveform viewer to show the simulation signals. The simulation results of the 10GBASE-R loopback design are as follows:

- 1. At 0 ns, the testbench drives the 156.25 MHz system clock to the DUT.
- 2. The MAC is released out of the reset. Signal O_CORE_RX_SRESET and O_CORE_RX_SRESET are at 0, as shown in the following figure.

Figure 13 • O_CORE_RX_SRESET and O_CORE_RX_SRESET at 0



3. The packet generator starts to send the packet. The start signal triggers the packet generation. The size of the packet is set to 0x32 (80 bytes). The sent packet can be viewed on the signals under the TX SIGNALS divider in the wave window.



Figure 14 • Ethernet Packet Sent



- 4. The packet checker receives the sent packet. The signals can be viewed under the RX SIGNALS divider in the wave window. The packet sent matches with the packet received.
- 5. The packet checker compares the incoming packet with the sent packet and increments the good packets (good_pckts) count by 1, as shown in the following figure.



Figure 15 • Good Packets Count Incremented by 1



The sent packet is looped back, and no errors are observed in the received packet, showing successful completion of 10GBASE-R Ethernet loopback.



3 Libero Design Flow

This chapter describes the Libero design flow, which involves the following processes:

- Synthesize, page 19
- Place and Route, page 20
- Verify Timing, page 21
- Generate FPGA Array Data, page 21
- Configure Design Initialization Data and Memories, page 22
- Generate Bitstream, page 22
- Run Program Action, page 23

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

Figure 16 • Libero Design Flow Options

<u>P</u> roject <u>F</u> ile <u>E</u> dit <u>V</u> iew Design Tools <u>H</u> elp
)esign Flow 🗗 🗙
Top Module(root): top 🖸 🚺 🌮
Tool
Netlist Viewer Synthesize Place and Route
Verify Post Layout Implementation
Verify Timing
🕰 Open SmartTime
🔍 🔍 Verify Power
Program and Debug Design
Generate FPGA Array Data
Configure Design Initialization Data and Memories
Generate Design Initialization Data
Configure Hardware
Device I/O States During Programming - ITAG Mode Only
Configure Programming Ontions
Configure Security
Program Design
V Generate Bitstream
Run PROGRAM Action
Program SPI Flash Image
Generate SPI Flash Image
Run PROGRAM_SPI_IMAGE Action
🖻 🕨 Debug Design
Identify Debug Design
😔 SmartDebug Design

3.1 Synthesize

To synthesize the design, perform the following steps:

1. On the Design Flow window, double-click Synthesize.

When the synthesis is successful, a green tick mark appears next to **Synthesize**, as shown in Figure 16, page 19.

2. Right-click **Synthesize** and select **View Report** to view the synthesis report and log files in the **Reports** tab. View the top_SD.srr and top_SD_compile_netlist.log files to debug synthesis and compile errors.



3.2 Place and Route

To place and route the design, TX_PLL, XCVR_REF_CLK, and PF_XCVR must be configured using the I/O Editor. Follow these steps to configure the components and place and route the design.

- 1. On the Design Flow window, double-click Manage Constraints.
- 2. On the I/O Attributes tab, click Edit with I/O Editor, as shown in the following figure.

Figure 17 • Edit with I/O Editor Option

st Eile Edit View Design Tools He

Project The Calc View Design Tools Trep	
Design Flow & X	So top* 5 × user.sdc 5 × top_derived_constraints.sdc 5 × Reports 5 × Retiming_Flops.v 5 × FiFo_wrappe
Top Module(root): top 🗈 🖸 🕼 🜮	/ 1/0 Attributes // Timing // Floor Planner // Netist Attributes /
Tool	New 🔻 Import Link Edit 👻 Check Help
Create Design	Edit with I/O Editor te
Create SmartDesign	constraint\io\usec.pdc [Target]
Create HDL	
Create SmartDesign Testbench	
Create HDL Testbench	
Verify Pre-Synthesized Design	
- Simulate	
E Constraints	
Manage Constraints	
Dr. Marta Vous	
S Suntharian	
W Discourse Dente	
A Verify Part I want Implementation	
Verify Post Layout Implementation	
Onen SmartTime	I/O secongs
Varify Dower	I Deserve Res for Desire Member
Program and Debug Design	I♥ Reserve Prins for Device Prigrauon
✓ • C Generate EPGA Array Data	Select the devices you are targetting for migration. Pins not bonded on these devices will be reserved in the device selected for this project.
Configure Design Initialization Data and Memories	Selected Devices IMP200TC EC - ECC1152
Generate Design Initialization Data	Selected Device. Mr 30015_3 - 1 03152
Configure Hardware	L MPF3001_ES
Programming Connectivity and Interface	Tarnat Davinari
- A Configure Programmer	larger bevices.
Device I/O States During Programming - JTAG Mode Only	
- 🗟 Configure Programming Options	
- 🔞 Configure Security	- General
😑 🕨 Program Design	
- 🤯 Generate Bitstream	V Reserve Pins for Probes
Run PROGRAM Action	
🖹 🖹 🕨 Program SPI Flash Image	
Design Flow Design Hierarchy Stimulus Hierarchy Catalog Files	

3. Using the **XCVR View** in I/O Editor, place TX_PLL, XCVR_REF_CLK, and PF_XCVR TX as shown in the following figure.

Figure 18 • I/O Editor Transceiver View



When all the components are placed, the location of the components is updated in the <code>user_fp.pdc</code> file (located in Constraint Manager > Floor planner tab), as shown in the following figure.



Figure 19 • Component Locations Updated in user_fp.pdc File

Reports	s & X StartPage & X Constraint Manager & X user_jo.pdc & X user_fp.pdc & X user.sdc & X
¢ 📃	<u>`</u>
1 []# Microsemi Physical design constraints file
2	L≢Using Quad2 Lanel XCVR for this design
3	set_location -inst_name PF_TX_PLL0_0/PF_TX_PLL0_0/txpll_isnt_0fixed_true -x 2460 -y 320
4	set location -inst name PF XCVR 0 0/I XCVR/LANE0 -fixed true -x 2460 -y 344
5	set location -inst name PF XCVR REFCLK 0/PF XCVR REFCLK 0/I IO -fixed true -x 2466 -y 317
6	

- On the Design Flow window, double-click Place and Route. When place and route is successful, a green tick mark appears next to Place and Route, as shown in Figure 16, page 19.
- 5. Right-click **Place and Route** and select **View Report** to view the place and route report and log files in the **Reports** tab. View the top_place_and_route_constraint_coverage.xml file for place and route constraint coverage.

3.2.1 Resource Utilization

The resource utilization report is written to the TOP_SD_layout_log.log file. To view this file, go to the **Reports** tab > **top reports** > **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.

Туре	Used	Total	Percentage
4LUT	4695	299544	1.57
DFF	4787	299544	1.60
I/O register	510	1536	0.00
Logic element	6134	299544	2.05

Table 5 •Resource Utilization

3.3 Verify Timing

To verify timing, perform the following steps:

- 1. On the **Design Flow** window, double-click **Verify Timing**.
 - When the design meets the timing requirements, a green tick mark appears next to **Verify Timing**, as shown in Figure 16, page 19.
- 2. Right-click **Verify Timing** and select **View Report** to view the verify timing report and log files in the **Reports** tab.

3.4 Generate FPGA Array Data

On the Design Flow tab, double-click Generate FPGA Array Data.

When the FPGA array data is generated, a green tick mark appears next to **Generate FPGA Array Data**, as shown in Figure 16, page 19.



3.5 Configure Design Initialization Data and Memories

The **Configure Design Initialization Data and Memories** option creates the non-PCIe transceiver initialization client, which initializes the transceiver block when the PolarFire device powers up.

To create the transceiver initialization client, perform the following steps:.

1. On the Design Flow window, double-click **Configure Design Initialization Data and Memories**. The Design and Memory Initialization window opens, as shown in the following figure.

Figure 20 • Design and Memory Initialization Window

Design	Flow	₽×	Reports 🗗 🗙	StartPage 🗗 🗙	Design and Memory Initialization 🖉 🗙
Тор	Module(root): TOP_SD 🗉 🖸 📳 :	Sr.	/ Design Initializatio		V SPI Flash V Fabric RAMs
		-	1		
	Tool		Apply	Discard	ep
	Create Design		In design initializat	ion, user design blocks	such as LSRAM, µSRAM, transceivers, and PCIe can be initialized as an option using data stored in the non-volatile storage memory.
	Create SmartDesign		The initialization d	ata can be stored in µP	ROM, sNVM, or an external SPI Flash.
	Create HDL		Follow the below r	tens to program the in	itialization data:
	🔐 Create SmartDesign Testbench		1. Set up your fat	ric RAMs initialization of	ata, if any, using the 'Fabric RAMs' tab
	Create HDL Testbench		2. Define the stor	age location of the initi	alization data
	E Verify Pre-Synthesized Design		3. Generate the in	itialization clients	
	Simulate		 Generate or ex Program the de 	port the bitstream	
	Constraints		of Program are de	vice	
	Manage Constraints		-Decise initializat	on consideration	
1	Implement Design		Congritinuoneur	on specification	
	🖓 Netlist Viewer		First stage (s	NVM)	
1	Synthesize				
1	Place and Route		In the first	stage, the initializa	tion sequence de-asserts FABRIC_POR_N.
1	Verify Post Layout Implementation				
1	💩 Verify Timing		Second stage	(sNVM)	
	🖞 Open SmartTime		In the seco	and stage, the initia	lization sequence initializes the PCTe and XCVR blocks present in the design.
	🖍 Verify Power				
	Program and Debug Design		Start addres	s for second stage initi	alization client: 0x 00000000
1	 Generate FPGA Array Data 				
	Configure Design Initialization Data and Memories		Third stage (PROM/sNVM/SPI-Flast)
V .	 Generate Design Initialization Data 		To the shire		Non-service to the Policy of the Policy of the design
	Configure Hardware		In the thin	i stage, the initializ	ation sequence initializes the rabic KAPIs present in the design.
L	Programming Connectivity and Interface		Memory type	e for third stage initializ	ation client: C uPROM
	- 👧 Configure Programmer				G ANIM
L	Device I/O States During Programming - JTAG Mode Only				
	Configure Programming Options				C External SPI-Flash (Non-authenticated) SPI Clock divider value: 1
	🔞 Configure Security		Charle address	- fee shiel as an initiali	where directs for another and
	🖶 🕨 Program Design		Start autres	s for a fird stage infrant	autor clience ox 100000000
V.	😽 🚱 Generate Bitstream				
	Run PROGRAM Action		Time Out (s):	128 🛨	
	Program SPI Flash Image				
	Generate SPI Flash Image		Custom configu	ration file:	
	Run PROGRAM_SPI_IMAGE Action				
	🖻 🕨 Debug Design				
	🔍 Identify Debug Design				
	- 🐵 SmartDebug Design				

- 2. Under **Second stage pane (sNVM)** enter the start address where the transceiver initialization client should be created in the sNVM, as shown in the preceding figure.
- 3. On the **Design Flow** window, double-click the **Generate Design Initialization Data** to generate the initialization client.

When the initialization client is generated, a green tick mark appears next to **Generate Design Ini-tialization Data**, as shown in Figure 16, page 19.

3.6 Generate Bitstream

To generate the bitstream, perform the following steps:

1. On the **Design Flow** tab, double-click **Generate Bitstream**.

When the bitstream is successfully generated, a green tick mark appears next to **Generate Bitstream**, as shown in Figure 16, page 19.

2. Right-click **Generate Bitstream** and select **View Report** to view the corresponding log file in the **Reports** tab.



3.7 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 as listed in the following table

Jumper	Description
J18, J19, J20, J21, and J22	Short pin 2 and 3 for programming the PolarFire FPGA through FTDI
J28	Short pin 1 and 2 for programming through the on-board FlashPro Express
J26	Short pin 1 and 2 for programming through the FTDI SPI
J27	Short pin 1 and 2 for programming through the FTDI SPI
J39	Short pin 1 and 2 for enabling the TX
J4	Short pin 1 and 2 for manual power switching using SW3

Table 6 • Jumper Settings for PolarFire Device Programming

- 2. Connect the power supply cable to the **J9** connector on the board.
- 3. Connect the host PC to the J5 connector (FTDI port) on the board using a USB cable.
- 4. Power on the board using the SW3 slide switch.
 - The following figure shows the PolarFire Evaluation Board setup for programming the device and running the reference design.

Figure 21 • PolarFire Evaluation Board Setup



 Double-click Run PROGRAM Action from the Libero > Design Flow tab. When the device is successfully programmed, a green tick mark appears next to Run PROGRAM Action, as shown in Figure 16, page 19.

For information about running the demo, see Running the Demo, page 24.



4 Running the Demo

Follow these steps to run the PolarFire 10GBASE-R loopback hardware demo design on the PolarFire Evaluation Board.

- 1. Insert a 10G SFP+ module into the J36 connector (SFP+ module cage) on the board.
- 2. Connect the J36 connector to the Spirent test module using an optical fiber cable.
- 3. On the host PC, start the Spirent TestCenter software.
- 4. In the Spirent TestCenter tree, under **All Ports**, click the port to which the optical fiber cable is connected. (In this demo, it is Port //1/2.)
- 5. On the **10Gig General** tab, select **Ethernet**.
- 6. Under **Ethernet Mode**, select **10G LAN**, and click **Apply**, as shown in the following figure. The attachment module details are automatically detected and populated.

Figure 22 • 10G Ethernet Settings

File Edit View Tools Actions Dia	agnostics Help	
🗋 💕 🛃 🎇 🔏 🐘 🛝 📾 Chassis 🗸	+] 🚵 Apply 😹 📇 🛱 🖏 🛯 🛱 🖏 00:00:00 Ķ % 🔐 📲 Technologies 📴 Perspective + 🔟 Sequencer 👔 Reporter 👔 Wizards + 🛕 Summary 🖉 Manage Tags 🖌 Manage Virtual Machines 👦	
Test Configuration		×
Grand Devices (brots, Roters,) Al Traffic Generators Al Traffic Generators Al Traffic Analyzers Al Traffic Analyzers O Al Ports Port///22 Devices Traffic Generator Traffic Generator Traffic Generator Traffic Generator Traffic Generator Storgs	Bit Presk Link & Restore link (B) Copy Wards TRGig Graveral 106ig Advanced FFC Measurement Loadion Port/line Sonet One One </th <th>^</th>	^
	Attachment Module Personality Board [file personality card file] Transceiver: 100BBASE-SR	*

After the changes are applied, the link status icon for the port turns green, indicating that communication has been established between the Spirent test module and the PolarFire Evaluation Board.

Figure 23 • Link Status

est Configuration	Break Link 🐁 Restore Link 👔	Copy Wizard FC Measurement Location	
All Traffic Generators	Port //1/2		Link Status 🥥
Al Traffic Analyzers Al Traffic Analyzers Al Traffic Analyzer Traffic Canalyzer Traffic Canalyzer Capture Settings	Ethernet Hode Ethernet Hode IoG LAN 106 VAN 106BASE-T 106BASE-T	SONET SONET Mode SPP G Gisco HDC SONET Line Speed 0 0C-3 0 C-12 0 0C-48 0 C-192	
	Attachment Module	ver 6 Transcriburg 100RASE-SR	



7. Click **Traffic Generator**, and in the **Add** list, click **Add Raw Stream Block**, as shown in the following figure.

Figure 24 • Stream Block Addition

	 Apply 	1 ER ER F	C E 1 E E E 00:00:08	🦻 👘 🔟 👔 lech	nologies	Perspective •	E Sequencer	Reporter	Re, Wizards	• 🛄 Summa	ry / Mana	ige lags 🤌	Manage Virtu	al Machines	E)
est Configuration															2
Al Device (host, Routers,) Al Traffic Generators Al Traffic Generators Al Traffic Analyzers Al Forts Port ///1/2 Port///1/2 Traffic Analyzer	Add - 😭 Add R Add R Po Lo	Generate S ound Stream aw Stream B rt Based ad per Stream anual Based	Iteram Block X Delete Brock(s) Block(s) Ith Utiliza m Block Schedule	F Edit P Copy W ion (%): 100 Burst Size: Inter Frame Gap: Inter Frame Gap Unit	1 12 : bytes	Duration Duration	n Mode: Cont	inuous v]						
Settings	Status	Active	Name	Tags	Index	Controlled By	Traffic Pattern	Туре	Tx Port	Rx Port	State	Load	Load Unit	Frame Length	iMIXDis
	_														
	_														_
	_														-
	_	-													
	_										-	-			-
															-

Select any one of the frame type from the options. Configure the frame size and frame format as shown in the following figure. Click **OK**.

Figure 25 • Stream Block Added

new.tcc - Spirent TestCenter			StreamBlock Editor - Port //10.60.138.42/1/2 : StreamBlock 3
File Edit View Tools Actions Di	iagnostics Help		General Groups Du Bart Droulou
👔 🚰 🛃 👫 🕹 🛝 🧰 Chassis - Test Configuration	• 濜 Apply 畿 尚 歸 時 鄧 略 0	0:00:15 🔏 😭 👔 Technologies 🛅 Perspective - 💽 Sequer	Active Name: StreamBlock 3
Al Device (rotes, Routers,) Al Traffic Generators Al Traffic Generator Al Traffic Generator	Add - Grensate Stream Block Scheduling Mode Bandwi Phrt Based Daad per Stream Block Manual Based Satus Active Name Active Name CreamBlock 3	Delete Caffi Copy Wizard	Frame size (Byteb)(With CRC and signature field) Street: Street:
w:Results 1	Change Result View Stream Blocks 1 - 1 Total Strees	eam Blocks: 1 Selected 1 of 1	Settings Padet Scheduling priority: 0 (b is the highest) 0 Burst sites: 1 Stat delay (bytes): 0
Basic Counters Errors Triggers Protocols Port Name Total Tx Count (Frames) (Frames)	Undersize/Oversize/Jumbo PFC Counters	User Defined Advanced Sequencing Generator CRC Error Count (Frames) Generator L3 Check	Navigate streamblods:
Port //10. L5L55.02.647 L5L55.92.6 2 2 1.51,55,00.141 L51,55.07.6 22: 1.51,55,00.141 L51,55.07.6	545 0 C		



8. Click **Port Load** (highlighted in blue in the Figure 25, page 25), and make sure that under **Fixed load settings**, the speed is set to 10000 Mbps, as shown in the following figure.Click **OK**.

Figure 26 • Port Load Settings

ad mode			
● Fix ○ Ran	idom		
ked load settings			
◯ Percent (%) :	100		
◯ Frame/sec (fps) :	14880952		
) bps :	1000000000		
) Kbps :	1000000		
Mbps :	10000	1	
) Inter burst gap (bytes) :	12	-	
) L2 Rate (bps):	7619047424		
			<u>о</u> к

- To begin Ethernet traffic generation, click the Start Traffic on all ports icon.
 The Port Traffic and Counters section starts displaying details of the Ethernet traffic:
 - The **Basic Traffic Results** section displays a real-time count of the Ethernet frames being transmitted and received. The number of Ethernet frames transmitted from the Spirent test module to the PolarFire Evaluation Board is displayed in the **Total Tx Count** field, and the number of Ethernet frames looped back by the PolarFire Evaluation Board to the Spirent test module is displayed in the **Total Rx Count** field.
 - The Aggregate Port Tx Rate section shows the rate at which the data is being transmitted.

Figure 27 • Ethernet Traffic Data

File Edit View Tools Actions	Diagnostics I	Help	1 191 1 191	P 00-00-18	1 🖉 🖉 💷 1 🛙	91 Techno	alogier 🔤 🗖	Perspective -	E Sequencer	@ Penorter	The Winards	- 🖪 Summ	ang I Z M	anage Tage	Manage Vir	tual Machiner	
Fest Configuration	Add •	🗱 Add - 📲 Generate Stream Block 🗙 Delete 🔐 Edt I 🌇 Copy Wizard 👘 Copy Wizard															
Al Devices (Hosts, Routers,) Al Traffic Generators Al Traffic Analyzers Al Traffic Analyzers Al Ports Al Ports Devices For Traffic Generator G Traffic Generator G Traffic Generator	Scheduling Mode Bandwidth Utiliz Bront Based Cad per Stream Block Manual Based Schedulic			lization (%): 100 Burst Size: Inter Frame (Inter Frame (In (%): 100 Burst Sze: I Duration Mode: Continuous Inter Frame Gap: 12 Inter Frame Gap Unit: bytes Advanced Port Load												
Sature Controlled by Traffic Pattern Type Tx Port Tx Port Traffic State Load					Load	Load Unit	Frame Lengt										
) e		StreamBlo	ck 2	c	Click to ad	0	generator	Pair	Port	Port //1/2	Any	2. Jup	Ready			Fixed
C Stram Block 1-1 Tetal Stram Block 1-1 Selected 1 of 1																	
asad:Results 1																	4 3
Port Traffic and Counters > Basic Traffic I	Results Chan	ge Result Vie	ew 🕶 🖹 🖁	5 🖪 I M -	1 of 1	Þ. Þ.		Port Traf	fic and Counters :	> Aggregate	Port L1 Tx R	ate Chang	e Result View	(• 6 <u>2</u> n →	12		
Basic Counters Errors Triggers Protoco	Total Pr Cou	Oversize/Jum	bo PFC Co	Px ECS Error	Defined Advance	iced Seque	ncing	Er.			Aggre	egate F	Port L1	Tx Rate)		
Port //1/2 / 25.105,654	14,999,755	anc (ridiles)		0	- reade count (Pr		0						Gbps				
Σ 15,109,884	14,999,755			0		0	0						10,00				
r							3	>									

As shown in the preceding figure, 10G Ethernet packets transmitted from the Spirent test module are successfully looped back by the PolarFire Evaluation Board.

10. Click Stop Generating Traffic 👫 icon to stop generating traffic from the Spirent test module.



5 Appendix 1: Programming the Device Using FlashPro Express

This chapter describes how to program the PolarFire device with the Job programming file using a FlashPro programmer. The default location of the Job file are located at the following location: mpf dg0757 df\Programming Files\DG0757 PF 10GBaseR.job

and

mpf dg0757 df\Programming Files\DG0757 PF 10GBaseR SYNCE.job

To program the PolarFire device using FlashPro Express, complete the following steps:

- 1. Ensure that the jumper settings on the board are the same as listed in Table 6, page 23.
- **Note:** The power supply switch must be switched off while making the jumper connections.
 - 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. On the host PC, launch the FlashPro Express software.
 - 6. To create a new job project, click New or
 - In the **Project** menu, select **New Job Project from FlashPro Express Job**, as shown in the following figure.



		FlashPro Express	
Job Projects		Project Edit View Programmer <u>H</u> elp	Ctrl+N
New Open		Open Job Project Close Job Project Save Job Project	Ctrl+O
Recent Projects	or	Set Log File Export Log File	Þ
		Execute Script Export Script File	Ctrl+U
		Recent Projects	•
		Exit	Ctrl+Q

- 7. 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: <download_folder>\mpf_dg0757_df\Programming_Files and <download_folder>mpf_dg0757_df\Programming_Files
- FlashPro Express job project location: Click Browse and navigate to the location where you want to save the project.



Figure 29 • New Job Project from FlashPro Express Job

😰 New Job Project from FlashPro Express Job	×
Programming job file: mpf_dg0757_liberosoc_v12_0_df\mpf_10GBaseR_df\Programming_File\PF_10FBASER.job	Browse
FlashPro Express job project name: PF_10FBASER FlashPro Express job project location:	
D:\Demo Design\Libero_v12p0_release\DG0757_PF_10GBaseR	Browse
Help	Cancel
or	
Project from FlashPro Express Job	×
Programming job file:	
$eq:mpf_dg0757_liberosoc_v12_0_df\mpf_SyncE_df\Programming_File\PF_10GBASER_SYNCE.job$	Browse
IashPro Express job project name: PF_10GBASER_SYNCE IashPro Express job project location:	
D:\Demo Design\Libero_v12n0_release\DG0757_PF_10GBaseRSvncF	Browse
louteeue e entruteeue - riche - oeaee (e eeue) - i - Teeeeeeeustuer	

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

Figure 30 • Programming the Device

Project E	dit View Program	nmer Help				
Refres	sh/Rescan Programm	ners				
ram	6 MPF300T Ф тво	() TDI 🗢				
PROGR	AM	•				
	RUN			IDLE		
Log				 	 	 ē ×
Messa	ages 🕹 Errors 🔺	Warnings 🕕 Info	,			



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

Figure 31 • FlashPro Express—RUN PASSED

Programmer Programmer Programmer Programmer PROGRAM PROGRAM PROGRAM PROGRAM PROGRAM PROGRAMMER(S) PASSED PROGRAMMER(S) PASSED
RUN I PROGRAM
RIGGRAM RUN RUN RUN RUN
ROGRAM I PROGRAMMER(S) PASSED
ROGRAM I PROGRAMMER(S) PASSED
RUN
INGRAM I PROGRAMMER(S) PASSED
NOGRAM I PROGRAMMER(S) PASSED
RUN 1 PROGRAMMER(S) PASSED
RUN 1 PROGRAMMER(S) PASSED
RUN 1 PROGRAMMER(S) PASSED
RUN 1 PROGRAMMER(S) PASSED
RUN I PROGRAMMER(3) PASSED
🛛 Messages 🔞 Errors 🔺 Warnings 🕕 Info
ANTARMAN IF2001DIV/2VI + Scan and Check Chain DAGGED
Ogrammer E2001RUKGY: device 'MPERGOT': Executing action PROGRAM
ogrammer 'E2001RUX6Y' : device 'MPF300T' : EXPORT ISC_ENABLE_RESULT[32] = 00000000
ogrammer 'E2001RUX6Y' : device 'MPF300T' : EXPORT CRCERR[1] = 0
ogrammer 'E2001RUX6Y' : device 'MPF300T' : Programming FPGA Array and sNVM
ogrammer 'E2001RUX6Y' : device 'MPF300T' : EXPORT BITS component bitstream digest[256] = e628346f15f57b0bb57f2a9a7ba39a841126caa067b4873fa03d218c2fd0001c
ogrammer 'E2001RUX6Y' : device 'MPF300T' : EXPORT Fabric component bitstream digest[256] = 45abf58f3e53ddee93e658b6c769ac04875a23b5394fe4a14beda9412919f5a5
ogrammer 'E2001RUX6%': device 'MPF300T': EXPORT sNVM component bitstream digest[256] = 79dfc8f8733ab3227al51990d92f5f1338fe8053c53c26cc76l6cd593c8fd566
ogrammer 'E2001R0X61' : device 'MFF3001' : EXFORT EOB component bitstream digest[256] = 2abf624bbe661/1aabd/94cd268668260b02ebbdd159F9F46/65/a6a/52/d8a0
ogrammer 'E2001R0Xer': device 'MPF3007':
Ogrammer 'E2001R0A61'; device 'MFF3001'; EXPORT DSN[120] = 6D/Ddd5522dd455455d25b26
Argenment (F2001DENES) - device (MDF2007) - Finished: Fri Feb 01 11:44:56 2019 (Flanged time 00:01:44)
Summary ESOCIONAL - device MESOCI - Executing action DACAM Discription
Grammer 12001BIXSY: Chain programming PASSED.
ain Frogramming Finished: Fri Feb UI 11:44:56 2019 (Klapsed Time UU:01:44)
ain Frogramming Finished: Fri Feb 01 11:44:56 2019 (Elapsed time 00:01:44)

11. Close FlashPro Express or in the Project tab, click Exit.



6 Appendix 2: Enabling SyncE in 10G BaseR Design

This section describes how to enable the SyncE Complaint Jitter Attenuation Phase Locked Loop (JA PLL) in the 10G BaseR Design. The design is validated using Optical Network Tester (ONT). The reference design describes how to build the 10G BaseR design with the JA PLL enabled.

6.1 **Prerequisite**

Before you begin:

- 1. Start Libero SoC, 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.
- 3. From the Project menu, click Open Project. The Open Project dialog box appears.
- 4. Navigate to the *mpf_dg0757_df\mpf_SyncE_df\Libero_Project* folder, select the Libero_Project.prjx file, and click **Open**.

The PolarFire 10G BaseR Ethernet SyncE reference design opens in the Libero.

5. Download the IP cores from Libero SoC Catalog.

The following figure shows the Hardware implementation of the 10G Base-R Ethernet SyncE loopback design.



Figure 32 • Hardware Implementation



6.2 Design Implementation

The following figure shows the Libero implementation of the 10G Base-R Ethernet loopback design with Jitter Attenuator PLL enabled.







6.2.1 Transceiver

To enable the Jitter attenuator DPLL, select the Enable JA_CLK in addition to the settings specified in Transceiver as shown in the following figure.

nsce	eiver Interface					
emicSy	rstemBuilder:PF_XCVR_ERM					
- 1	PMA Settings					<u> </u>
(C)	[]X data rate	10312.5	Mbps	BX data rate	10312.5 Mbps	
BA BA	TX dock division factor	1	•	RX CDR lock mode	Lock to data	
All M	IX PLL base data rate	10312.500	Mbps	BX CDR reference dock source	Dedicated	
IR IR	TX PLL bit clock frequency	5156.250	MHz	RX CDR reference dock frequency	156.25 MHz	1
IR I	PCS Settings	_				
IR IR	TX PCS-Fabric interface width	32	▼ bits	BX PCS-Pabric interface width	32 v bits	
1 R 3G	TX FPGA interface frequency	322.266	MHz	RX FPGA interface frequency	322.266 MHz	
HC SD	BMA Mode Frashle CDR Bit-dia port					
lal Ial	36 10b Encoding/Decoding					 PE XCVR 0
	5 64b6xb Gear Box					
				⊆ 64667b		-CTILLOX UND
	Enable Disparity			Enable BER monitor state machin	e	ICAL/MERCIPLES LANE/ACCIMUM
	Enable Scrambler/Descrambler			Enable 32 bits data width		AMELING A UNDERCONTA UNDERCONTA
	Soft PIPE Interface					
	Protocol	PCIe Gen1 (2.5 Gbps)	<u></u>			AMELINGAN AND A AMELING AMELINGAN AND A AMELINGAN AND A AMELING AMELINGAN AND AMELINGAN AND AMELING AMELINGAN AND AMELINGAN AND AMELING AMELINGAN AND AMELINGAN AND AMELINGAN AND AMELING AMELINGAN AND AMELINGAN AND AMELINGA
	Clocks and Resets					LANE_TO CONTROL
	Interface Clocks					 PF_XCVR
_	Use as PLL reference dock					
	IX dock	Global	· 0	BX dock	Regional	
	Interface Resets					
	BMA Reset	TX and RX	v	<u>PCS</u> Reset	TX and RX	
	Optional Ports			Finable BX_BEADY_CDB and BX	VAL CDR ports	
	Enable TX ELEC IDLE port			Finable JA CLK port		
	Dynamic Reconfiguration					
	Switch between two TX PLLs	(DKI)		Switch between two CDR referen	nce dodis	
4						Symbol
essages	🛿 😂 Errors 🔺 Warnings 🕕 Info					
-						



6.2.1.1 Transmit PLL

The clock option is set to Jitter cleaning mode to enable the Jitter attenuator PLL as shown in the following figure.

Figure 35 • Transmit PLL—SyncE

Configurator	-		×
Transmit PLL Microsemi:SgCore:PF_TX_PLL:2.0.002			
Transmit PLL			
Clock Inputs/Outputs Reference Clock Source Dedicated 156.25 MHz			
Desired Output Bt Clock [103122.500 Mbps 5156.250 MHz PF_	TX_PLL	_0	
-REF_CLK -JA_REF_CLK	CLKS_	PLL_LOCK _TO_XCVR	
Cosk Options PF	_TX_PL	.L	
G 3tter Cleaning Mode L0G SyncE 328t			
Feabures Frable Dynamic Reconfiguration Interface (DRI)			
4 Symbol			
Help 🔻	ок	Canci	

The remaining IP configurations, clocking structure, and the reset structure of this design is same as the demo design explained earlier. For more information, see Demo Design, page 4.

6.3 Libero Design Flow

For more information, see Libero Design Flow, page 19.

6.4 Programming the Device Using FlashPro Express

The .job programming file is programmed using a FlashPro Express programmer. The file is located at:

mpf dg0757 df\Programming Files

For more information, see Appendix 1: Programming the Device Using FlashPro Express, page 27.



6.5 Running the Demo

Follow these steps to run the PolarFire 10GBASE-R loopback hardware demo design on the PolarFire Evaluation Board.

 The reference design is validated using the ONT. There exists the ppm offset in the TX and RX frequencies when the Jitter cleaning Mode is not enabled in the TX PLL as shown in the following figure.

Figure 36 • Frequency Offset—SyncE Disabled

VIAVI ONT-6 Mo	600 Location: ONT- odule E 10G/2.5G H-0009 Port 1 Application: Us	-606 DA-235 Slot 1.1 10.60.138.30 Module Time: 12:18 PM IST ser-Application Disk: 4.8GB of 7.5GB free
All Layers OK Rx Port: XFP Interface Pow. Ovd. LOS Pow. Ovd. Erro Interface Stat. Stat. Stat. Stat. Erro Help Optical Power:	YS Interface Payload XFP Info SFP 1 Info S TX Bitrate: 10.313 Gb/s [10GigE] Port: XFP [43] Wavelength: 850 nm	RX Bitrate: 10.313 Gb/s [10GigE] Port: XFP [43] Optical Power: 10.5 LOS 0verload -15 -12 -1 +2
Frequency: 10.312.614 HHz Frequ. Offset: 111 ppm	Frequency Offset 0.0 ppm Transition Ramp Clock Source: Internal [from Clock Module]	Frequency Offset: Out of Range Frequency: 10,312,614 kHz
P	[31,32]	
🗲 Insertion 🛛 💾 🤞	🔼 Laser	Elapsed: 00d 00h 00m 00s of Continuous Start



2. When the design is built with Jitter Cleaning Mode of TX PLL enabled the Frequency offset between Tx and Rx is 0 ppm.

Figure 37 • Frequency Offset Between TX and RX

VIAVI	ONT-600 Module I	Location: ONT-606 10G/2.5G H-0009 Port 1 Application: User-Ap	DA-235 Slot 1.1 10.60.138.30 pplication	Module Time: 12:08 PM IST Disk: 4.8GB of 7.5GB free
Current Alarms /	PHYS Config.	Interface Payload XFP Info SFP 1 Info SFP 2	P Info RX	
Errors	Status Overview	Bitrate: 10.313 Gb/s [10GigE]	Bitrate: 10.313 Gb/s [10)GigE]
Rx Port: XFP	Alarms / Errors	Port: XFP [43]	Port: XFP [43]	▼
Dev. Ovid. Pow. Ovid. Freq. Rng. Optical Power:	Help	Wavelength: 850 nm	Optical Power: LOS Overload -15 -12	-4.2 dBm -1 +2
Frequency: 10,312,500 kHz Frequ. Offset: 0.0 ppm	MAG	Frequency Offset 0.0 ppm Transition Ramp	Frequency Offset:	0.0 ppm +2'00
			Frequency: 10,312,50) kHz
		Ref. Clock Output: High Range 644.53 MHz [51,52]		
		Co Laser		
			Default	
🗲 Insertion	🗅 🛕 Lase	u C	Elapsed: 00d 00h 00m 00s of Con	inuous 🔻 Start

When the Tx clock frequency is offset by 100 ppm, the Rx Clock frequency also gets adjusted by 3. 100 ppm, which shows the JA PLL is tracking and adjusting the clock as per the offset in the received Clock.

Figure 38 • Frequency Offset—SyncE Enabled

VIAVI	ONT-600 Module	Locativ E 10G/2.5G H-0009 Port 1 Applic:	on: ONT-606 DA-235 Slot 1.1 10.60.138.30 ation: User-Application	Module Time: 12:12 PM IST Disk: 4.8GB of 7.5GB free
All Layers OK R. Port: XFP Interface V Pow. Ovid. Freq. Rng. Optica Pow.	Help	Interface Payload XFP Info SFP 1 In TX Bitrate: 10.313 Gb/s (10GigE) Port Port XFP [43] Image: State Stat	To SFP 2 Info	s [10GigE]
Frequency: 10.313.531 HHz Frequ. Offset 100.0 ppm	MA	Frequency Offset 100.0 ppm Transition	n Ramp Frequency Offset: Out of Range -200 Frequency: 10,312	+100.0 ppm +200
	L	Ref. Clock Output: High Range 644.53		



7 Appendix 3: References

This section lists documents that provide more information about the concepts and features covered in this demo guide.

- For more information about 10G Ethernet, refer to the IEEE 802.3 standard in the IEEE website.
- For more information about PolarFire transceiver blocks, see UG0677: PolarFire FPGA Transceiver User Guide.
- For more information about Libero, ModelSim, and Synplify, visit Microsemi Libero SoC webpage.