DG0799
Demo Guide
PolarFire FPGA 1G Ethernet Loopback Using IOD CDR

Microsemi
a Microchip company

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

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

About Microsemi

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

Contents

Figures

Tables

1 Revision History

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

1.1 Revision 5.0

Added Appendix 2: Running the TCL Script, page 28.

1.2 Revision 4.0

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.

1.3 Revision 3.0

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

- Updated the document for Libero SoC v12.1.
- The design uses a new IP PF_IOD_CDR_CCC. For more information, see PF_IOD_CDR_CCC_C0, page 10.

1.4 Revision 2.0

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

- Updated the document for Libero® SoC v12.0.
- Added Appendix 3: Multi-Lane 1G IOD CDR Design, page 29.
- Added Appendix 4: 1G Ethernet BASE-T and BASE-X Using Transceiver, page 31.

1.5 Revision 1.0

The first publication of this document.

2 PolarFire FPGA 1G Ethernet Loopback Using IOD CDR

Microsemi PolarFire® FPGAs support 1G (1000BASE-T) Ethernet solutions for various networking applications. In PolarFire devices, 10/100/1000 Mbps (1G) Ethernet is implemented using the CoreTSE Media Access Control (MAC) soft IP core. The CoreTSE IP implements a Serial Gigabit Media-Independent Interface (SGMII) with an Ethernet PHY. This Ethernet interface can be implemented in the FPGA by using either a transceiver or a GPIO with Clock and Data Recovery (CDR) capability. Both these features are provided by the PF_XCVR and PF_IOD_CDR IP cores, respectively.

GPIOs in PolarFire devices operate at speeds of up to 1.066 Gbps for single-ended standards and 1.25 Gbps for differential standards. Each I/O has an I/O digital (IOD) logic block that supports gearing up of the output data rate and gearing down of the input data rate. The IOD block with CDR circuitry (PF_IOD_CDR IP) deserializes high-speed Ethernet input data and transfers it to the FPGA fabric at lower speeds. It also serializes the lower-speed Ethernet data from the FPGA fabric and transfers to the high-speed Ethernet PHY.

This document describes how to run the 1G Ethernet loopback demo design, which is a reference design created to demonstrate 1G Ethernet loopback using GPIO on a PolarFire Evaluation Board. The demo design is built using the PF_IOD_CDR_CCC, PF_IOD_CDR, CoreTSE, and Mi-V soft processor IP cores. The reference design is for a single SGMII lane (single RJ45 cable). For information about how to build a multi-lane (multiple links) design, see Appendix 3: Multi-Lane 1G IOD CDR Design, page 29.

The demo design can be programmed using either of the following options:

- Using the pre-generated .job file: To program the device using the .job file provided along with the demo design files, see Appendix 1: Programming the Device Using FlashPro Express, page 26.
- Using Libero SoC: To program the device using Libero SoC, see Libero Design Flow, page 15.

A license is required to use the CoreTSE IP core. To request a license, contact *soc_marketing@microsemi.com*.

2.1 Design Requirements

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

Table 1 • **Design Requirements**

Requirement	Version
Hardware	
PolarFire Evaluation Kit (POLARFIRE-EVAL-KIT) – PolarFire Evaluation Board – 12 V/5 A AC power adapter and cord – USB 2.0 A to mini-B cable for UART and programming	Rev D
RJ45 cable to connect the board with the host PC	
Host PC	Windows 7 or 10
Software	
Cat Karat (Ethernet packet generator)	To install the appropriate version, refer *PacketBuilder website*. **Note:** The Cat Karat version that was used in this demo is: 1.51.200
Wireshark (network protocol analyzer)	To install the appropriate version, refer *Wireshark website*. **Note:** The Wireshark version that was used in this demo is: 3.4.3
FlashPro Express	**Note:** Refer to the `readme.txt` file provided in the design files for the software versions used with this reference design.
Libero SoC Design Suite	

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:

1. For demo design files download link:

   http://soc.microsemi.com/download/rsc/?f=mpf_dg0799_eval_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
3. The latest versions of ModelSim and Synplify Pro are included in the Libero SoC PolarFire installation package. Make sure you have a Libero Gold license for design evaluation on MPF300 device. A one year Gold software License is included with the Evaluation kit.
4. If you already purchased a Gold license and received a Software ID from Microsemi, generate your Gold License using the following link:
   https://soc.microsemi.com/portal/default.aspx?r=1
5. Download Cat Karat and Wireshark.

Note: The Cat Karat version 1.51.200 and Wireshark version 3.4.3 were used in this demo.

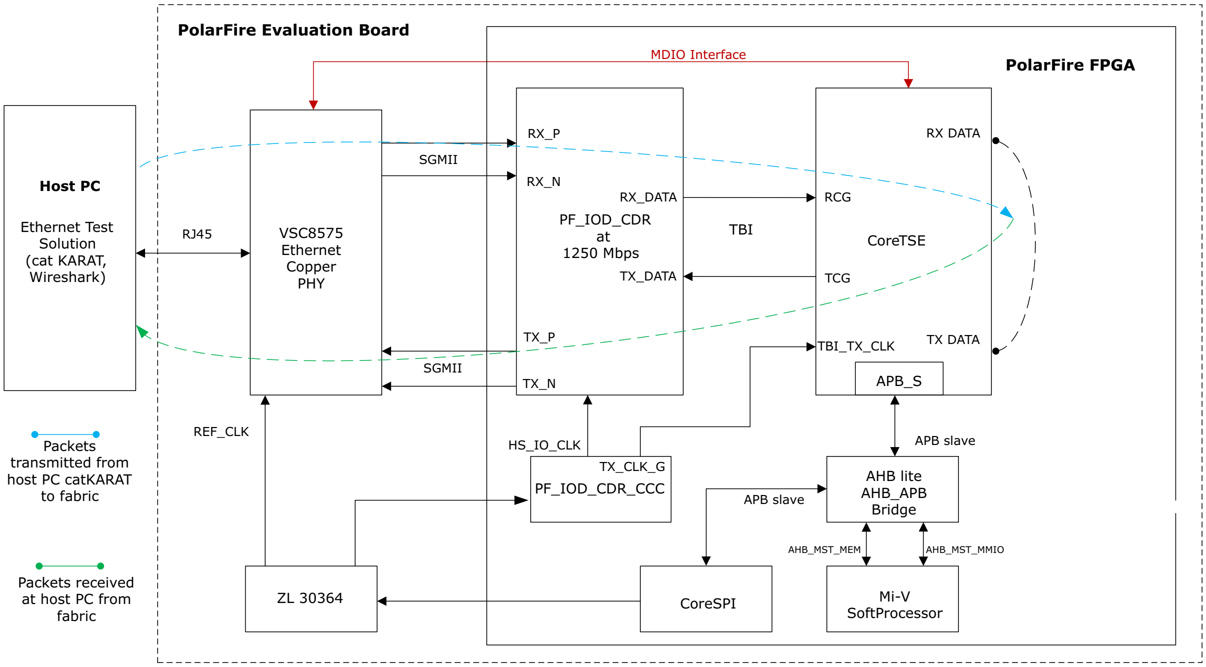
2.3 Demo Design

The following is the data flow for the 1G Ethernet loopback demo design:

1. PF_CCC_0 provides the clock to the Mi-V processor and other APB peripherals.
2. PF_IOD_CDR_CCC_C0 generates:
   - The fabric transmit clock ((TX_CLK_G)) for the CoreTSE block.
   - The high-speed bank clocks, and drives the high-speed clocks (HS_IO_CLKs) of the PF_IOD_CDR_C0 block for clock recovery.
3. PF_IOD_CDR_CCC_C0 also generates Delay codes for the PVT compensation.
4. Mi-V performs the following functions:
   - Executes the application from TCM.
   - Configures the ZL30364 clock generation hardware through the CoreSPI IP to generate reference clocks for the VSC PHY and the IOD CDR fabric module.
   - Configures the Management registers of CoreTSE and VSC PHY.
   - Sends a request to the CoreTSE IP to negotiate with the on-board VSC8575 PHY.
5. CoreTSE IP implements 1G Ethernet MAC and is configured in ten bit interface mode (TBI) to interface with the PF_IOD_CDR_C0. The CoreTSE IP has an inbuilt MDIO interface to exchange control and status information with the VSC PHY.
6. PF_IOD_CDR IP does the following:
   - Interfaces with the on-board VSC8575 PHY and forms the SGMII link.
   - Recovers the data and clock from the incoming RX_P and RX_N ports.
   - Sends the recovered clock (RX_CLK_R) to the CoreTSE block.
   - Deserializes the recovered data and sends 10-bit parallel data to CoreTSE.
   - Receives Ethernet data through the RX_P and RX_N input pads, gears down the receive data rate, and deserializes the data.
   - The deserialized data is sent from PF_IOD_CDR_C0:RX_DATA [9:0] to CoreTSE IP: RCG [9:0].
   - The received data is looped back at the CoreTSE IP, and CoreTSE IP:TCG[9:0] is sent to PF_IOD_CDR_C0:TX_DATA [9:0].
   - PF_IOD_CDR_C0 serializes the data, gears up the transmit data rate, and transmits the data to the on-board VSC PHY through the TX_P and TX_N output pads.

Figure 1 shows the hardware implementation of the demo design.

Figure 1 • **Block Diagram**



2.3.1 About PF_IOD_CDR

The PF_IOD_CDR IP core provides an asynchronous receive and transmit interface that supports up to 1.6 Gbps speed for serial data transfers. It supports the SGMII interface. PF_IOD_CDR uses the DDRX5 IO gearing mode for the SGMII interface with a 10:1 digital ratio to provide the 10-bit data width for both transmit and receive. The clock recovery circuit, which is part of this PF_IOD_CDR, keeps the receive clock centered in the data eye.

The PF_IOD_CDR interface is compatible with the CoreTSE, CoreTSE_AHB, and CoreSGMII IP cores configured in TBI mode. In this demo, the CoreTSE (Non-AMBA) MAC is used in the TBI mode to transmit and receive the Ethernet packets.

2.3.1.1 Receive interface

The PF_IOD_CDR IP includes the clock recovery block, which is used to generate the recovered clock for sampling the incoming data stream. This IP uses the four clocks of phases 0, 90,180, and 270 degrees for the clock recovery. The recovered clock (RX_CLK_R) is used by the fabric for sampling the Rx data from the PF_IOD_CDR IP. The CoreTSE logic also uses this clock.

For more information PF_IOD_CDR and its blocks, see *UG0686: PolarFire FPGA User I/O User Guide*.

2.3.1.2 Transmit Interface

The PF_IOD_CDR transmit interface receives the parallel data (TX_DATA [9:0]), converts it into a serial data stream using the IOD interface, and then transmits it through the I/O ports TX_P and TX_N. The 625 MHz clock generated by the PF_IOD_CDR_CCC is used by PF_IOD_CDR transmit interface to transmit the data serially on the TX_P/TX_N ports.

For more information about PF_IOD_CDR, see *UG0686: PolarFire FPGA User I/O User Guide.*

2.3.2 Design Implementation

Figure 2 shows the top-level Libero implementation of the demo design.

Figure 2 • **Top-Level Libero Implementation**

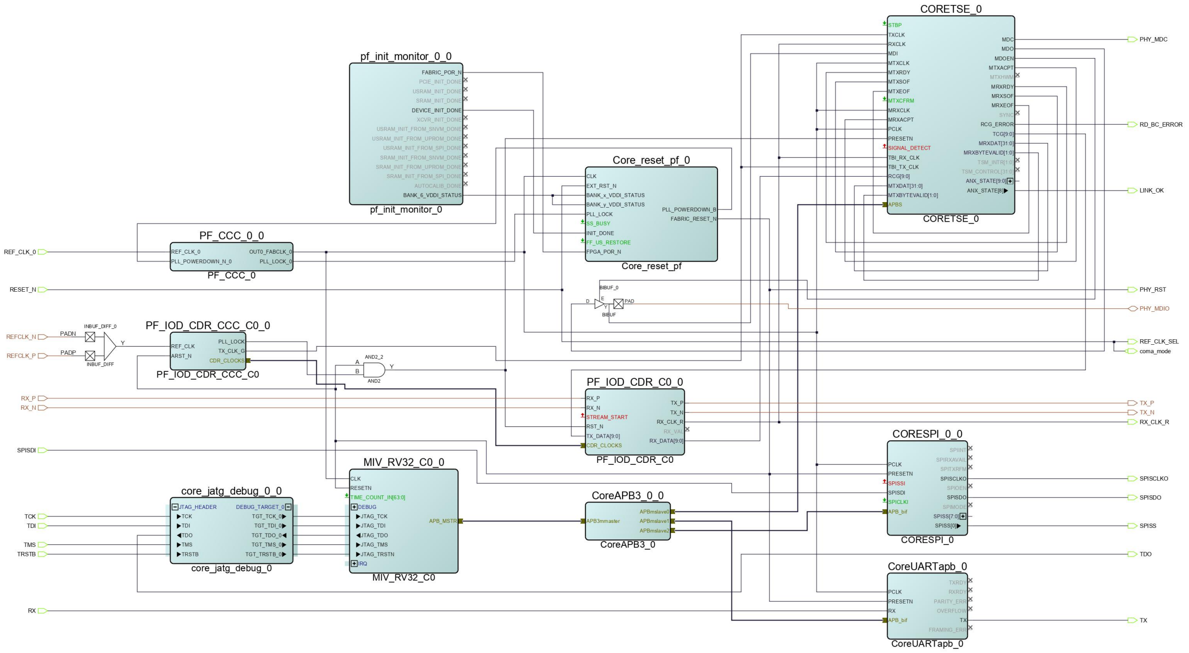


Table 2 lists the important I/O signals of the design.

Table 2 • **I/O Signals**

Signal	Direction	Description
RX_P, RX_N	Input	IOD CDR receive signals connected to the VSC PHY transmit data signals
TX_P, TX_N	Output	IOD CDR transmit signals connected to the VSC PHY receive data signals
REFCLK_N, REFCLK_P	Input	125 MHz input clock received from the on-board ZL30364 and fed to NWC_PLL_0
RESET_N	Input	Active low Mi-V reset. Asserted by pressing the on-board K22 push-button
REF_CLK_0	Input	50 MHz input clock received from the on-board 50 MHz oscillator and fed to PF_CCC_0
TCK, TDI, TMS, TRSTB	Input	JTAG signals interfaced to the soft processor for debugging
LINK_OK	Output	Link status indicator. Provides the link *up* or *down* status with the on-board PHY. This signal is mapped to on-board LED7. The LED ON condition indicates that the link is up.
PHY_RST	Output	Active high reset signal to the on-board VSC8575 PHY
PHY_MDC	Output	Management Data IO clock fed to the on-board VSC8575 PHY
PHY_MDIO	Output	Management Data IO Interface for accessing the on-board VSC8575 PHY registers
coma_mode	Output	Signal held low to keep the VSC PHY fully active when it is out of reset
REF_CLK_SEL	Output	Reference clock speed pin of the VSC PHY. Held high for selecting the 125 MHz reference clock speed
RD_BC_ERROR	Output	CoreTSE receive error signal. This LED signal indicates the receive code group error. This signal is synchronous to RX_CLK_R and mapped to LED4 on the board. When the LED is ON, there is an error in the received code group. When the LED is OFF, there is no error.
SPISCLKO, SPISS, SPISDO, SPISDI	Output	SPI controller signals to interface with the ZL30364 clock generation hardware
TDO	Output	JTAG test data output. Serial data output from tap.

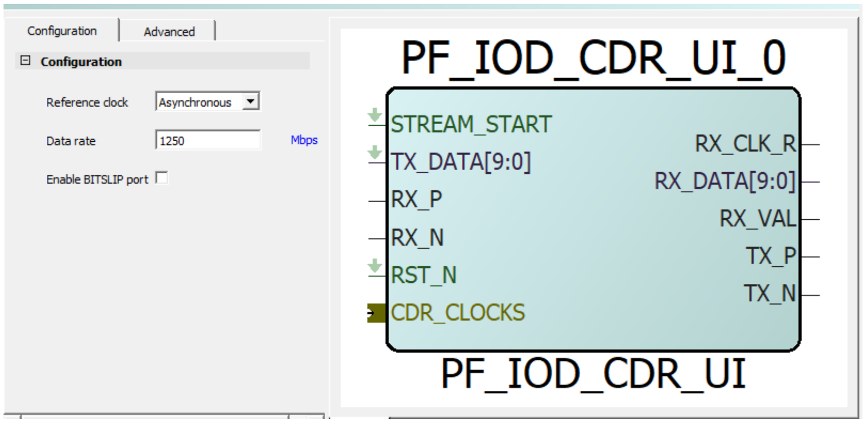
2.3.3 IP Configuration

This section describes the IP blocks and user-defined blocks instantiated in the demo design.

2.3.3.1 PF_IOD_CDR_C0_0

The PF_IOD_CDR_C0_0 (PF_IOD_CDR) block is configured for 1250 Mbps. The data rate is set to 1250 Mbps because the SGMII interface operates at this speed. The **Enable BITSLIP port** check box is not selected because the CoreTSE IP has a built-in word alignment logic.

Figure 3 • **PF_IOD_CDR Configurator**



Configuration | Advanced

⊟ Configuration

Reference clock: Asynchronous

Data rate: 1250 Mbps

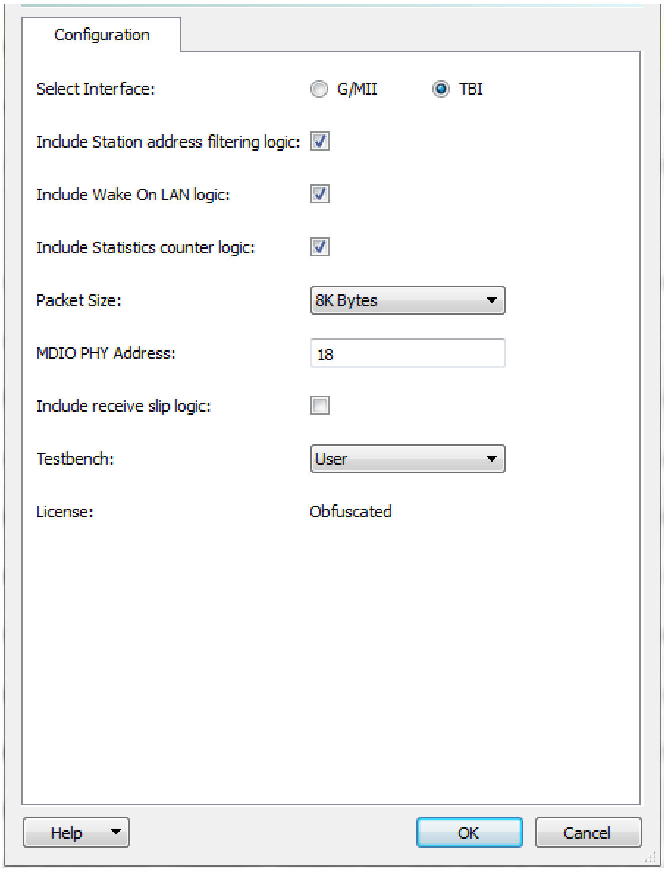
Enable BITSLIP port □

The Advanced tab includes the Jump step size option that specifies the precision of the clock adjustment during clock recovery. The supported step sizes are 2 or 3, this demo uses a step size of 3. Figure 3 shows the configuration of the PF_IOD_CDR_C0 block.

2.3.3.2 CORETSE_0

The CORETSE_0 (CoreTSE) block is used to implement the Ethernet MAC. This block is configured in the ten-bit interface (TBI) mode to interface with the VSC PHY using the SGMII interface, as shown in Figure 4. The **MDIO PHY Address** value is used by the Mi-V soft processor to read and write to the Management registers of the CoreTSE IP. The Include receive slip logic option is not selected because the CoreTSE IP has a built-in word alignment logic in TBI mode.

Figure 4 • **CORETSE_0 Configurator**



Select Interface: ○ G/MII ● TBI

Include Station address filtering logic: ☑

Include Wake On LAN logic: ☑

Include Statistics counter logic: ☑

Packet Size: 8K Bytes

MDIO PHY Address: 18

Include receive slip logic: □

Testbench: User

License: Obfuscated

2.3.3.3 pf_init_monitor_0

The pf_init_monitor_0 (PF_INIT_MONITOR) block is used to issue a reset signal to the user logic (FABRIC_RESET_N). To ensure a glitch-free reset, the DEVICE_INIT_DONE signal is connected to the CORERESET_PF IP with a lock signal from the PF_CCC macro.

This IP retains the default configuration.

2.3.3.4 Core_reset_pf_0

The Core_reset_pf_0 (CORERESET_PF) block handles the sequencing of reset signals in the PolarFire device. The CORERESET_PF block synchronizes the reset of all the blocks to which it is connected when the PolarFire device is powered up.

2.3.3.5 core_jtag_debug_0

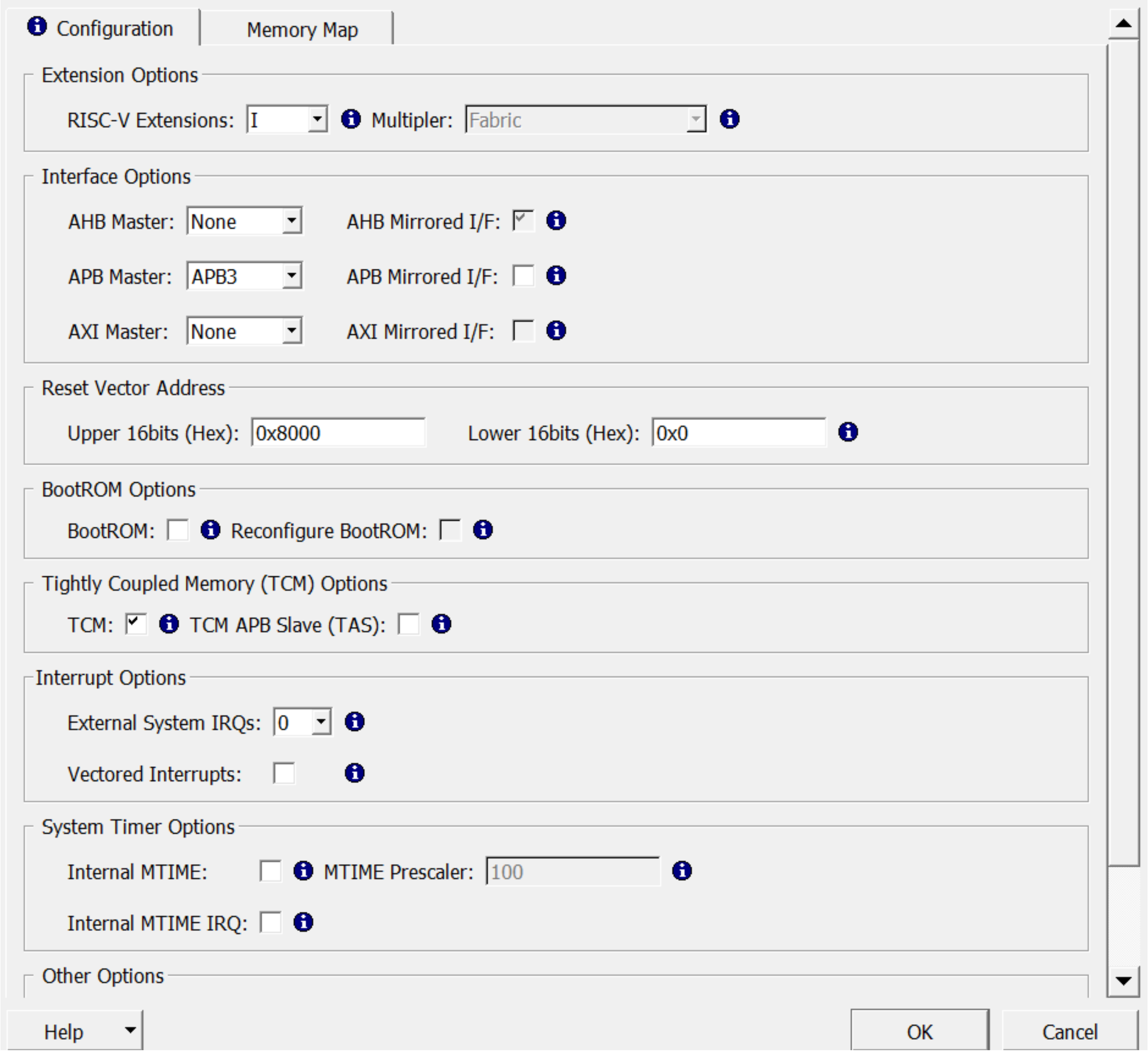
The CoreJTAGDebug IP is used to debug the Mi-V soft processor. This IP retains the default configuration.

2.3.3.6 Mi-V Soft Processor

The Mi-V soft processor supports RISC-V processor-based designs. It configures the ZL30364 clock generation hardware through the CoreSPI IP and the VSC PHY through the CoreTSE MDIO interface. It also configures the CoreTSE registers using the APB interface.

The following figure shows the Mi-V soft processor configuration, where the **Reset Vector Address** is set to 0x8000_0000. This is because in the Mi-V processor memory map, the memory range used for the APB interface is 0x6000_0000 to 0x6FFF_FFFF, and the memory range used for TCM is 0x8000_0000 to 0x8000_FFFF.

Figure 5 • **Mi-V Configurator**



Extension Options

RISC-V Extensions: I Multipler: Fabric

Interface Options

AHB Master: None AHB Mirrored I/F: ☑

APB Master: APB3 APB Mirrored I/F: □

AXI Master: None AXI Mirrored I/F: □

Reset Vector Address

Upper 16bits (Hex): 0x8000 Lower 16bits (Hex): 0x0

BootROM Options

BootROM: □ Reconfigure BootROM: □

Tightly Coupled Memory (TCM) Options

TCM: ☑ TCM APB Slave (TAS): □

Interrupt Options

External System IRQs: 0

Vectored Interrupts: □

System Timer Options

Internal MTIME: □ MTIME Prescaler: 100

Internal MTIME IRQ: □

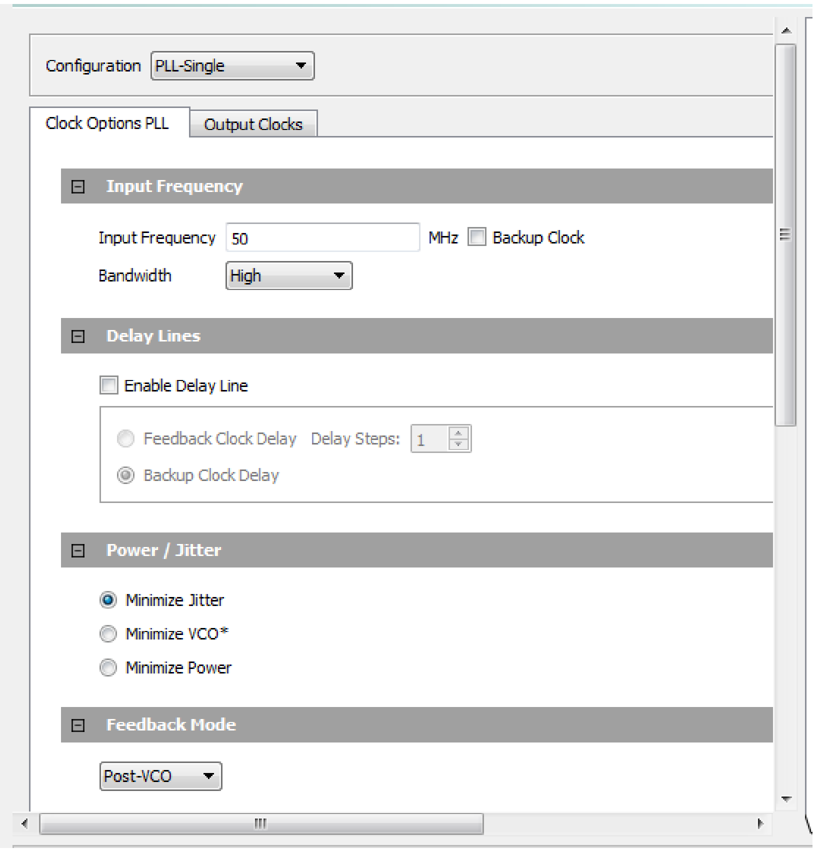
Other Options

2.3.3.7 PF_CCC_0

The PF_CCC_0 (PolarFire Clock Conditioning Circuitry) generates the fabric reference clock that drives the soft processor and the APB peripherals (CoreTSE and CoreSPI). The PF_CCC_0 IP is configured to generate one output fabric clock from an on-board 50 MHz crystal oscillator.

Figure 6 shows the PF_CCC_0 input clock configuration.

Figure 6 • **PF_CCC_0 Input Clock Configuration**


Configuration: PLL-Single

Clock Options PLL | Output Clocks

Input Frequency

Input Frequency: 50 MHz ☐ Backup Clock

Bandwidth: High

Delay Lines

☐ Enable Delay Line

○ Feedback Clock Delay   Delay Steps: 1

● Backup Clock Delay

Power / Jitter

● Minimize Jitter

○ Minimize VCO*

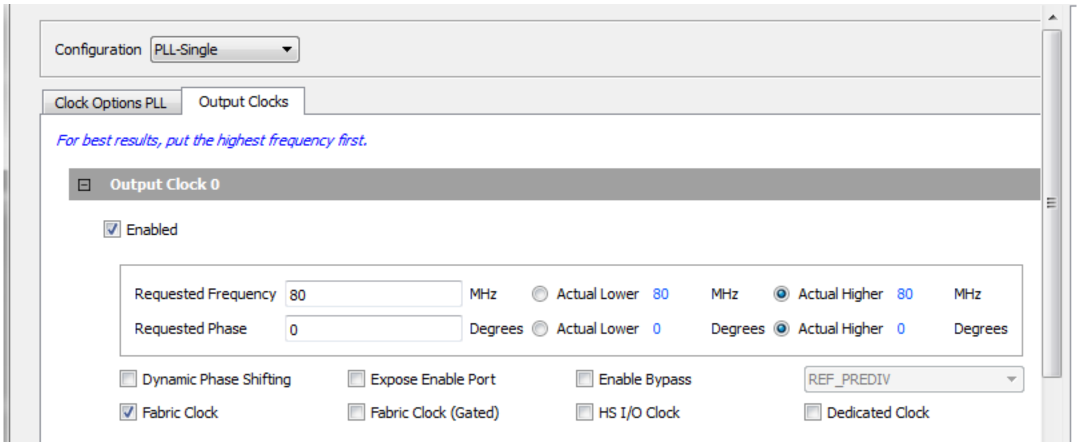
○ Minimize Power

Feedback Mode

Post-VCO

[Figure 7](#) shows the PF_CCC_0 output clock configuration. This design uses an 80 MHz system clock for configuring the APB peripherals.

Figure 7 • **PF_CCC_0 Output Clock Configuration**


Configuration: PLL-Single

Clock Options PLL | Output Clocks

For best results, put the highest frequency first.

Output Clock 0

☑ Enabled

Requested Frequency	80	MHz	○ Actual Lower 80	MHz	● Actual Higher 80	MHz
Requested Phase	0	Degrees	○ Actual Lower 0	Degrees	● Actual Higher 0	Degrees

☐ Dynamic Phase Shifting   ☐ Expose Enable Port   ☐ Enable Bypass   REF_PREDIV

☑ Fabric Clock   ☐ Fabric Clock (Gated)   ☐ HS I/O Clock   ☐ Dedicated Clock

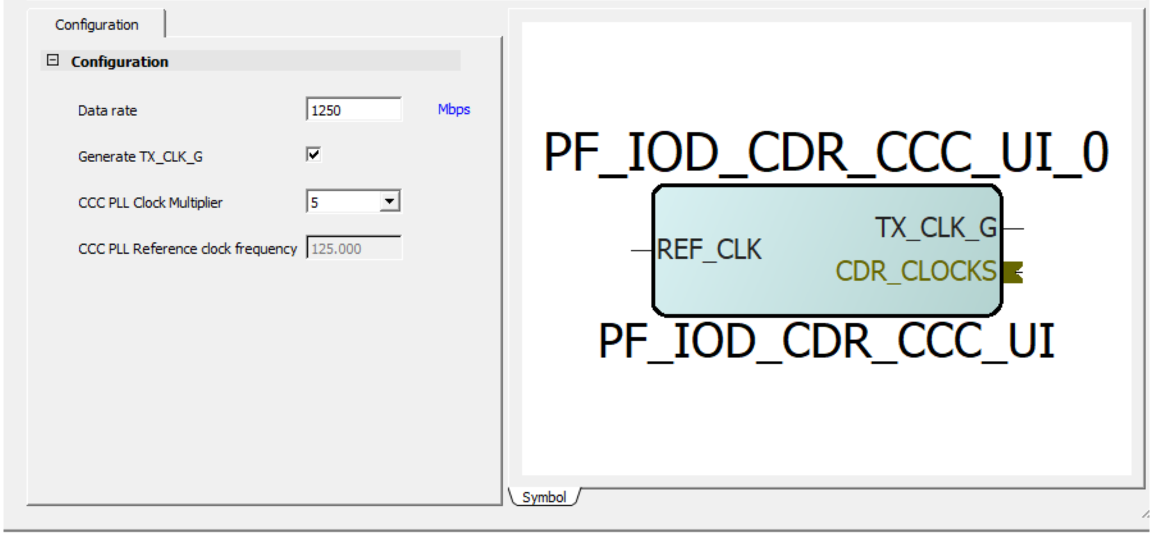
2.3.3.8 PF_IOD_CDR_CCC_C0

The PF_IOD_CDR_CCC IP used for generating high-speed bank clocks for PF_IOD_CDR. This IP is in PLL-DLL cascaded mode to generate high-speed bank clocks of four phases 0, 90, 180, 270 from a 125 MHz input. The CDR requires four phases of the HS_IO_CLK running at half the frequency of the serial data rate. Therefore, the HSIO clock frequency is selected as 625 MHz with four phases.

PF_IOD_CDR_CCC also generates the fabric Tx interface clock (TX_CLK_G) for the CoreTSE block by dividing the bank clock by the ratio of 5. The DLL is needed to control the clock position (delay) with DLL codes when the data is active on the Rx interface (RX_P/RX_N). A glitch-less DLL can adjust the clock delay setting when the data is active.

Figure 8 shows the PF_IOD_CCC_C0_0 configuration.

Figure 8 • **PF_IOD_CDR_CCC Configuration**



Configuration

⊟ Configuration

Data rate 1250 Mbps

Generate TX_CLK_G ☑

CCC PLL Clock Multiplier 5

CCC PLL Reference clock frequency 125.000

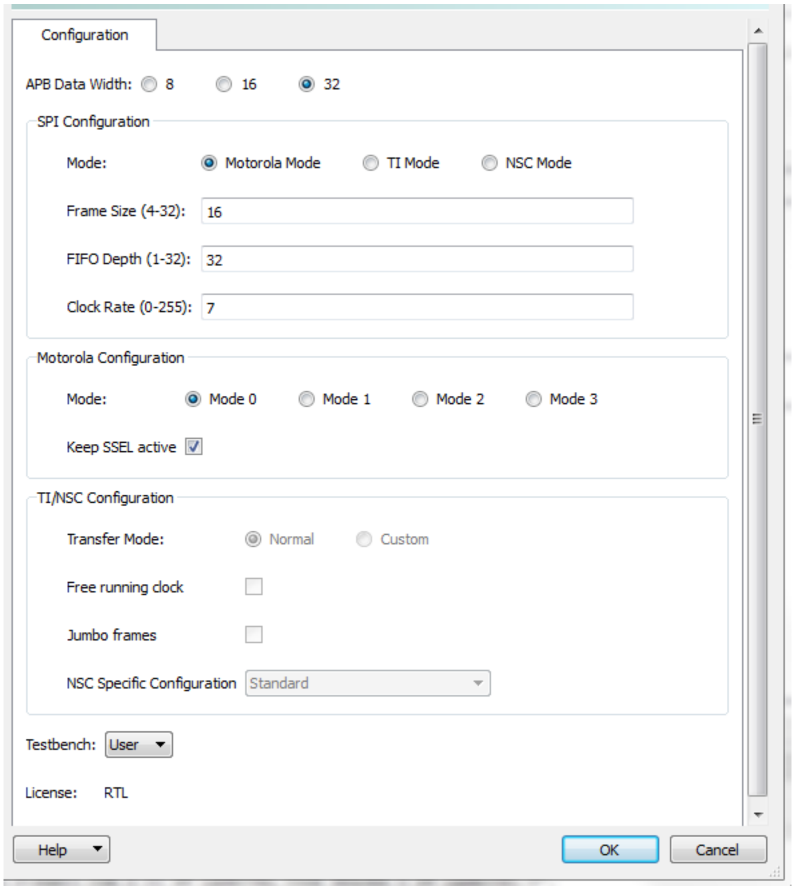
2.3.3.9 CORESPI_0

The CORESPI_0 (CoreSPI) block is a controller IP, which implements SPI communication. Mi-V configures the ZL30364 clock generation hardware using the CORESPI_0 block. The following points describe the CoreSPI_0 configuration, as shown in Figure 9.

- APB Data Width is selected as 32 because the design uses an APB data width of 32 bit.
- The default serial protocol mode, Motorola mode is retained to interface with ZL30364.
- Frame size is set to 16 to match the read/write cycles supported by ZL30364.
- FIFO depth is set to 32 to store maximum frames (TX and RX) in FIFO.
- Clock rate for the SPI master clock is selected as 7. This is used to generate the SPI clock of 5 MHz (SPICLK = PCLK/(2*(clock rate+1) = 80/(2*(7+1)) = 5 MHz).
- The **Keep SSEL active** check box is enabled to keep the slave peripheral active between back-to-back data transfers.

Figure 9 shows the CoreSPI configuration.

Figure 9 • **CoreSPI_0 Configurator**



APB Data Width: ○ 8 ○ 16 ● 32

SPI Configuration

Mode: ● Motorola Mode ○ TI Mode ○ NSC Mode

Frame Size (4-32): 16

FIFO Depth (1-32): 32

Clock Rate (0-255): 7

Motorola Configuration

Mode: ● Mode 0 ○ Mode 1 ○ Mode 2 ○ Mode 3

Keep SSEL active ☑

TI/NSC Configuration

Transfer Mode: ● Normal ○ Custom

Free running clock □

Jumbo frames □

NSC Specific Configuration Standard

Testbench: User

License: RTL

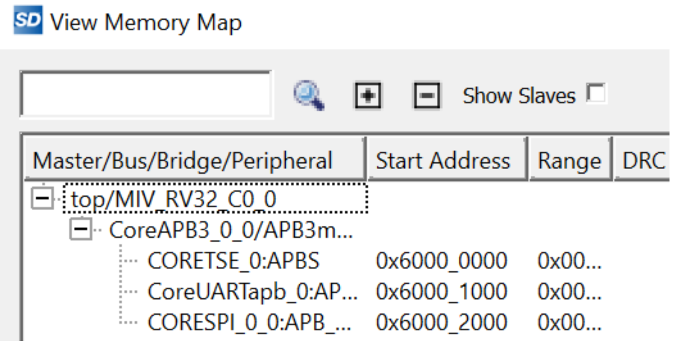
2.3.3.10 CoreUARTapb_0

The CoreUARTapb_0 block is a serial communication controller with a flexible serial data interface. It is used for UART communication between the device and host PC. This IP retains the default configuration.

2.3.3.11 Design Memory Map

Figure 10 shows the Mi-V processor bus interface memory map.

Figure 10 • **Mi-V Processor Bus Interface Memory Map**



View Memory Map

Show Slaves □

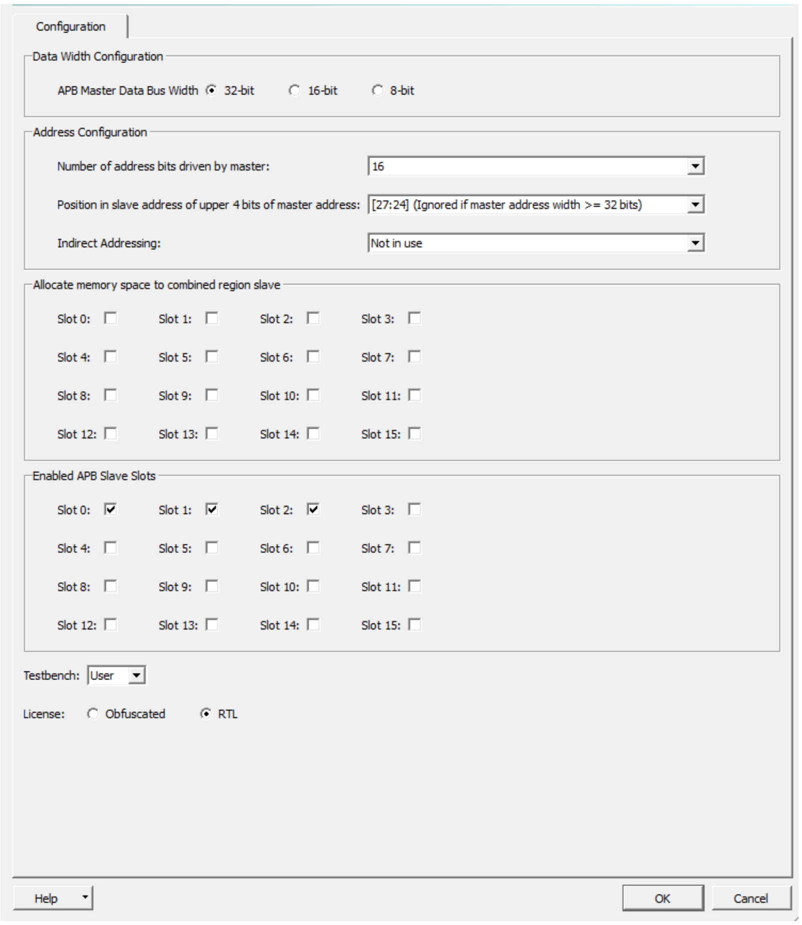
Master/Bus/Bridge/Peripheral	Start Address	Range	DRC
top/MIV_RV32_C0_0			
CoreAPB3_0_0/APB3m...			
CORETSE_0:APBS	0x6000_0000	0x00...	
CoreUARTapb_0:AP...	0x6000_1000	0x00...	
CORESPI_0_0:APB_...	0x6000_2000	0x00...	

2.3.3.12 CoreAPB3

CoreAPB3 is configured as shown in Figure 11 to connect the peripherals CoreTSE, CoreSPI, and CoreUARTapb as slaves.

- APB Master Data bus width: 32 bit
- Number of address bits driven by master: 16. The Mi-V processor addresses slaves using 16-bit addressing, so the final address for these slaves translates to 0x6000_0000, 0x6000_1000, and 0x6000_2000
- Enabled APB Slave Slots: S0, S1, and S2 (for CoreTSE, CoreUARTapb, and CoreSPI, respectively).

Figure 11 • **CoreAPB3 Configuration**



Configuration

Data Width Configuration

APB Master Data Bus Width ● 32-bit ○ 16-bit ○ 8-bit

Address Configuration

Number of address bits driven by master: 16

Position in slave address of upper 4 bits of master address: [27:24] (Ignored if master address width >= 32 bits)

Indirect Addressing: Not in use

Allocate memory space to combined region slave

Slot 0: □ Slot 1: □ Slot 2: □ Slot 3: □
Slot 4: □ Slot 5: □ Slot 6: □ Slot 7: □
Slot 8: □ Slot 9: □ Slot 10: □ Slot 11: □
Slot 12: □ Slot 13: □ Slot 14: □ Slot 15: □

Enabled APB Slave Slots

Slot 0: ☑ Slot 1: ☑ Slot 2: ☑ Slot 3: □
Slot 4: □ Slot 5: □ Slot 6: □ Slot 7: □
Slot 8: □ Slot 9: □ Slot 10: □ Slot 11: □
Slot 12: □ Slot 13: □ Slot 14: □ Slot 15: □

Testbench: User

License: ○ Obfuscated ● RTL

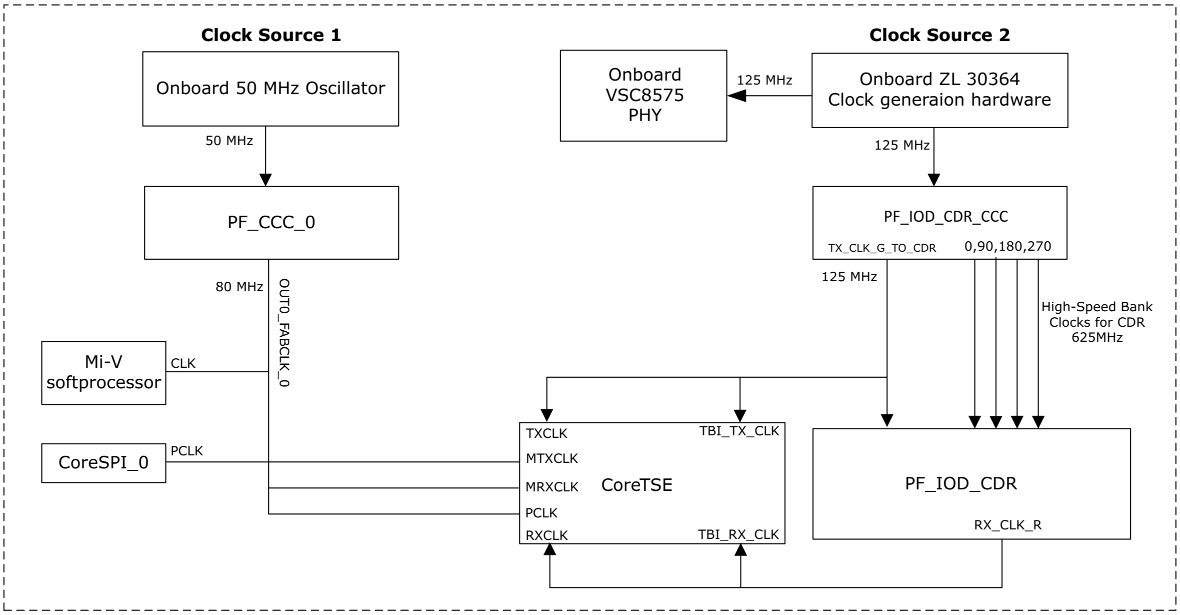
2.4 Clocking Structure

In the demo design, there are two clock sources—the on-board 50 MHz oscillator and the on-board ZL30364 clock generation hardware.

- **On-board 50 MHz oscillator:** This oscillator drives the PLL that generates an 80-MHz clock for the Mi-V soft processor and peripherals. In this design, Mi-V processor runs at 80 MHz.
- **On-board ZL 30364 clock generation hardware:** This hardware generates the reference clocks for the VSC PHY, the IOD CDR fabric module, and CoreTSE.

Figure 12 shows the clocking structure of the demo design.

Figure 12 • **Clocking Structure**



3 Libero Design Flow

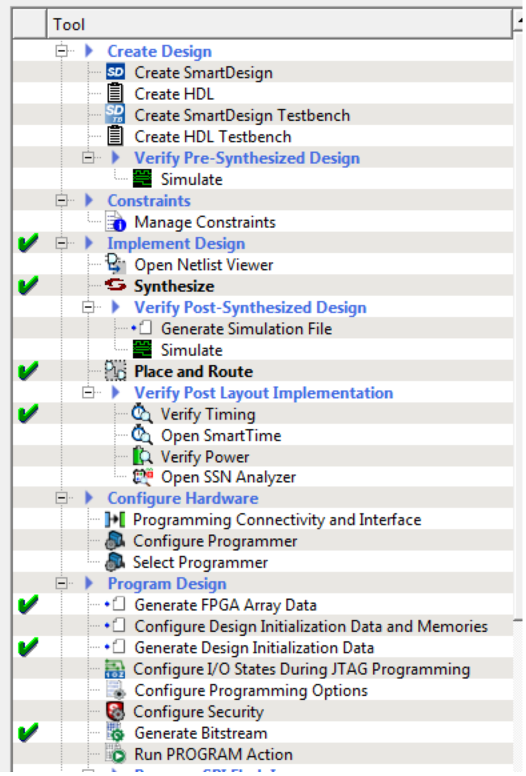
This chapter describes the Libero design flow for running this demo design, which includes:

- Synthesize, page 15
- Place and Route, page 16
- Verify Timing, page 16
- Generate FPGA Array Data, page 16
- Configure Design Initialization Data and Memories, page 17
- Generate Bitstream, page 18
- Run PROGRAM Action, page 18

Note: To initialize the TCM in PolarFire using the system controller, a local parameter **l_cfg_hard_tcm0_en**, in the `miv_rv32_opsrv_cfg_pkg.v` file should be changed to 1'b1 prior to synthesis. See the 2.7 TCM section in the *MIV_RV32 Handbook*.

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

Figure 13 • **Libero Design Flow Options**



3.1 Synthesize

To synthesize the design, perform the following the 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 13.
2. Right-click **Synthesize** and select **View Report** to view the synthesis report and log files in the **Reports** tab.
   We recommend viewing the `top.srr` and `top_compile_netlist.log` files for debugging synthesis and compile errors.

3.2 Place and Route

The demo project includes the IO PDC file and the floor planner PDC constraint files. The Place and Route process uses these PDC files to place the I/Os and CCC macros.

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

1. 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 13, page 15.
2. Right-click **Place and Route** and select **View Report** to view the place and route report and the log files in the **Reports** tab.
   We recommend viewing the `top_place_and_route_constraint_coverage.xml` file for place and route constraint coverage.

3.2.1 PLL, DLL, and Lane Controller Placement

PolarFire FPGA I/O pairs are grouped into lanes. Each I/O bank has multiple lanes. Each lane consists of twelve I/Os (six I/O pairs), a lane controller, and a set of high-speed, low-skew clock resources.

All associated I/Os must be placed in one lane. For example, RX_P and RX_N must be placed in the same lane. For more information, see *UG0686: PolarFire FPGA User I/O User Guide*. The IO Editor shows the placement of the components and I/Os. The PLL of PF_CCC and the PLL, DLL, and Lane Controller of PF_IOD_CDR_CCC_C0 are auto placed by Libero SoC.

3.2.2 Resource Utilization

The resource utilization report is written to the `top_layout_log.log` file in the **Reports** tab under iog_cdr_1 Gbps reports > Place and Route. Table 3 lists the resource utilization of the design after place and route. These values may vary slightly for different Libero runs, settings, and seed values.

***Table 3* • Resource Utilization**

Type	Used	Total	Percentage
4LUT	22551	299544	7.53
DFF	12703	299544	4.24
I/O register	0	510	0.00
Logic Elements	24090	299544	8.04
User I/O	21	512	4.10
– Single-ended I/O	17	512	3.32
– Differential I/O pairs	3	256	1.17

3.3 Verify Timing

To verify timing, perform the following steps:

1. On the **Design Flow** window**,** double-click **Verify Timing**.
   When the design successfully meets the timing requirements, a green tick mark appears next to **Verify Timing**, as shown in Figure 13, page 15.
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** window, double-click **Generate FPGA Array Data**.

When the FPGA array data is successfully generated, a green tick mark appears next to **Generate FPGA Array Data**, as shown in Figure 13, page 15.

3.5 Configure Design Initialization Data and Memories

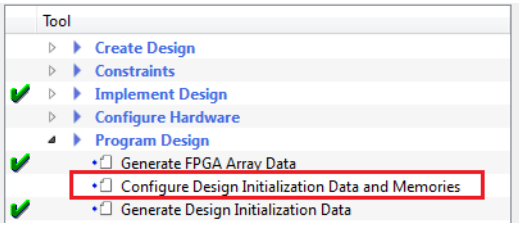
The fabric RAM blocks must be initialized with the user application to configure the PHY and management registers of CoreTSE. The user application (HEX file) is generated using SoftConsole. This step is used to select the fabric RAM client (HEX file), its storage location (sNVM/µPROM/SPI Flash), and generate the fabric RAM client. The non-volatile memory is programmed with this client and at device power-up, the fabric RAM blocks are initialized with the content from the selected NVM.

The Configure Design Initialization Data and Memories option creates the TCM initialization client. When the PolarFire device powers up, the TCM memory is initialized with the sNVM contents.

To create the TCM initialization client, perform the following steps:

1. On the **Design Flow** window, double-click **Configure Design Initialization Data and Memories**, as shown in Figure 14.

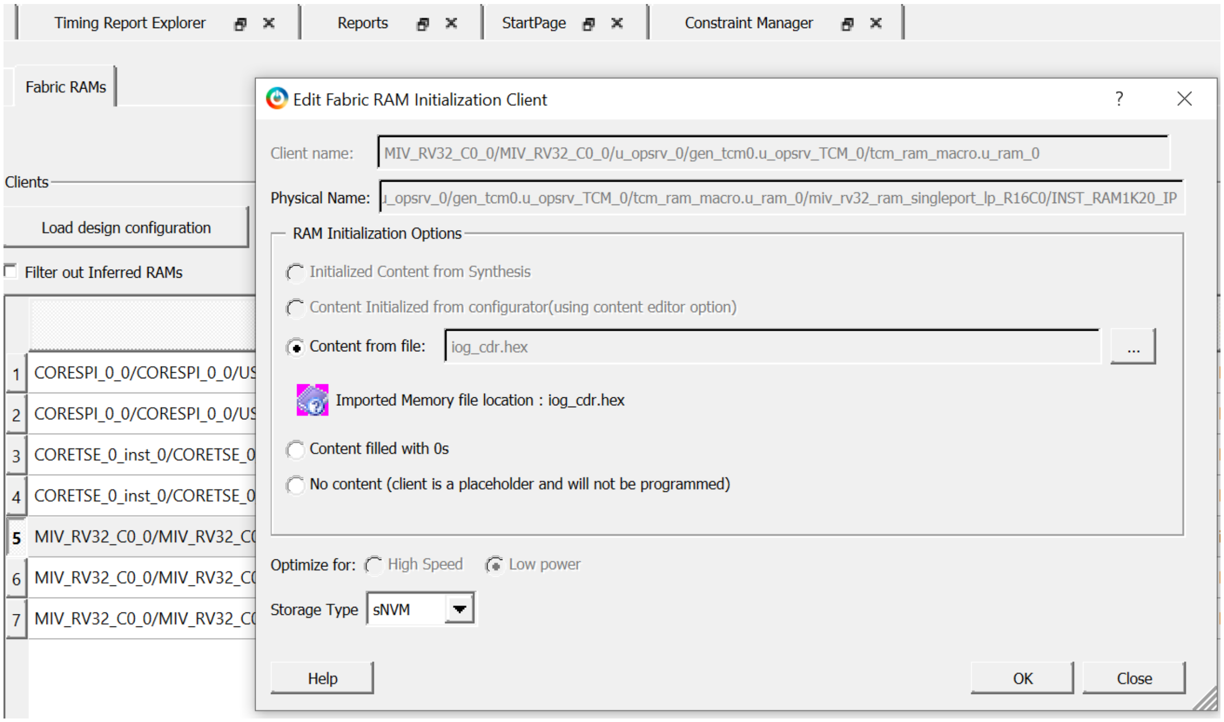
Figure 14 • **Configure Design Initialization Data and Memories Option**



Tool
- Create Design
- Constraints
- ✓ Implement Design
- Configure Hardware
- Program Design
  - ✓ Generate FPGA Array Data
  - Configure Design Initialization Data and Memories
  - ✓ Generate Design Initialization Data

2. In the Fabric RAMs tab, configure the MIV_RV32_C0_0/MIV_RV32_C0_0/u_opsrv_0/gen_tcm0.u_opsrv_TCM_0/tcm_ram_macro.u_ram_0 instance and ensure contents are stored in sNVM, set the Storage Type as sNVM as shown in Figure 15.

Figure 15 • **Fabric RAMs Tab**



☐ Filter out Inferred RAMs

1	CORESPI_0_0/CORESPI_0_0/US
2	CORESPI_0_0/CORESPI_0_0/US
3	CORETSE_0_inst_0/CORETSE_0
4	CORETSE_0_inst_0/CORETSE_0
5	MIV_RV32_C0_0/MIV_RV32_C0
6	MIV_RV32_C0_0/MIV_RV32_C0
7	MIV_RV32_C0_0/MIV_RV32_C0

Edit Fabric RAM Initialization Client

Client name: MIV_RV32_C0_0/MIV_RV32_C0_0/u_opsrv_0/gen_tcm0.u_opsrv_TCM_0/tcm_ram_macro.u_ram_0

Physical Name: u_opsrv_0/gen_tcm0.u_opsrv_TCM_0/tcm_ram_macro.u_ram_0/miv_rv32_ram_singleport_lp_R16C0/INST_RAM1K20_IP

RAM Initialization Options
- ○ Initialized Content from Synthesis
- ○ Content Initialized from configurator(using content editor option)
- ● Content from file: iog_cdr.hex
  Imported Memory file location : iog_cdr.hex
- ○ Content filled with 0s
- ○ No content (client is a placeholder and will not be programmed)

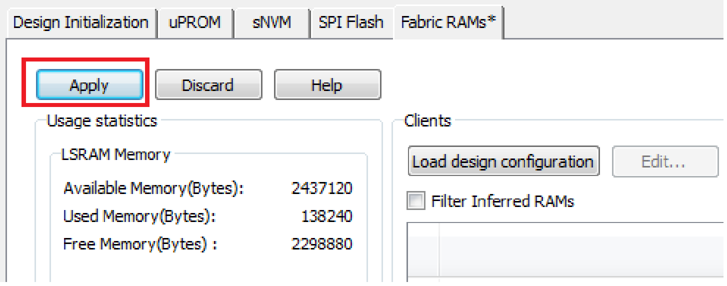
Optimize for: ○ High Speed ● Low power

Storage Type: sNVM

3. Import the hex file (`iog_cdr.hex`) from: `mpf_dg0799_eval_df\Libero_Project`.
   The `iog_cdr.hex` file is an application file generated using SoftConsole v6.5 that configures the ZL clock generation hardware, the CoreTSE registers, and the VSC PHY. The application code is initially stored in sNVM. On device power-up, the system controller copies the code to TCM, and the Mi-V processor executes the code from TCM.

4. Click **Apply,** as shown in Figure 16.

Figure 16 • **Fabric RAM Tab Apply Option**



Available Memory(Bytes): 2437120
Used Memory(Bytes): 138240
Free Memory(Bytes) : 2298880

5. On the **Design Flow** window, double-click **Generate Design Initialization Data**.
   When the initialization client is successfully generated in sNVM, a green tick mark appears next to Generate Design Initialization Data, as shown in Figure 13.

3.6 Generate Bitstream

To generate the bitstream, perform the following steps:

1. On the **Design Flow** window, double-click **Generate Bitstream**.
   When the bitstream is successfully generated, a green tick mark appears next to **Generate Bitstream**, as shown in Figure 13, page 15.
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. The programming procedure involves setting up board and invoking the programming command from Libero.

Follow these steps:

Note: If you want to program the board using the .job file instead, see Programming the Device Using FlashPro, page 26.

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

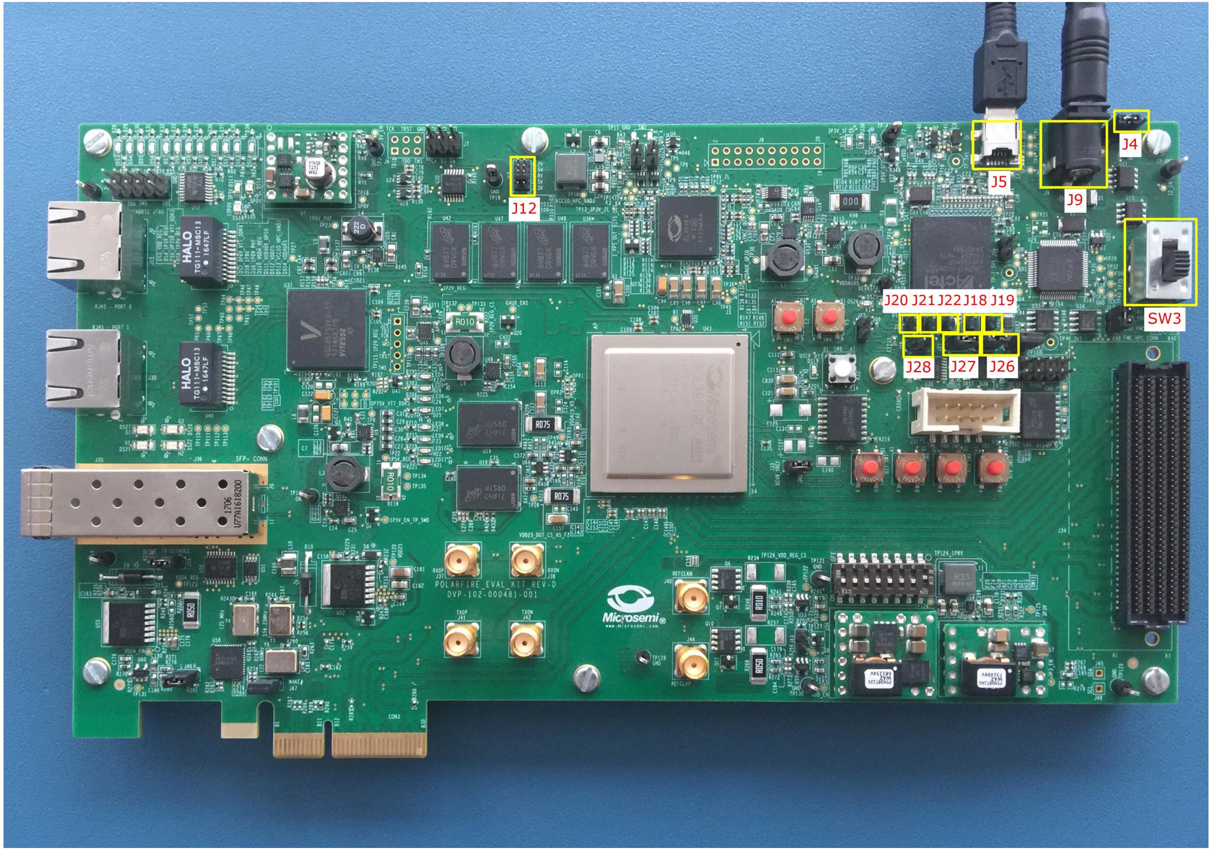
Table 4 • **Jumper Settings**

Jumper	Setting
J18, J19, J20, J21, and J22	Close pins 2 and 3 for programming through FTDI
J28	Close pins 1 and 2 for programming through the on-board FlashPro5
J4	Close pins 1 and 2 for switching the power manually using SW3
J12	Close pins 3 and 4 for 2.5 V

2. Connect the power supply cable to the **J9** connector on the board.
3. Connect the USB cable from the host PC to **J5** (FTDI port) on the board.
4. Connect the RJ45 cable from the host PC to the **J15** connector (RJ45-PORT 0) on the board. This is required for the Ethernet link after programming.

The following figure shows the board setup for programming the device.

Figure 17 • **Board Setup**



5. Power up the board using the **SW3** slide switch.
6. On the **Design Flow** window, double-click **Run PROGRAM Action**.
   When the device is successfully programmed, the LEDs 6, 7, 8, 9, 10, and 11 on the board glow, and a green tick mark appears, as shown in Figure 13, page 15.
7. Right-click **Run Program Action** and select **View Report** to view the corresponding log file in the **Reports** tab.

The demo is ready to be run. For information about how to run the demo, see Running the Demo, page 20.

4 Running the Demo

This section describes how to run the 1G loopback demo. The procedure involves transmitting packets from the network card of the host PC to the board using Cat Karat and verifying the packets transmitted to and received from the board using Wireshark.

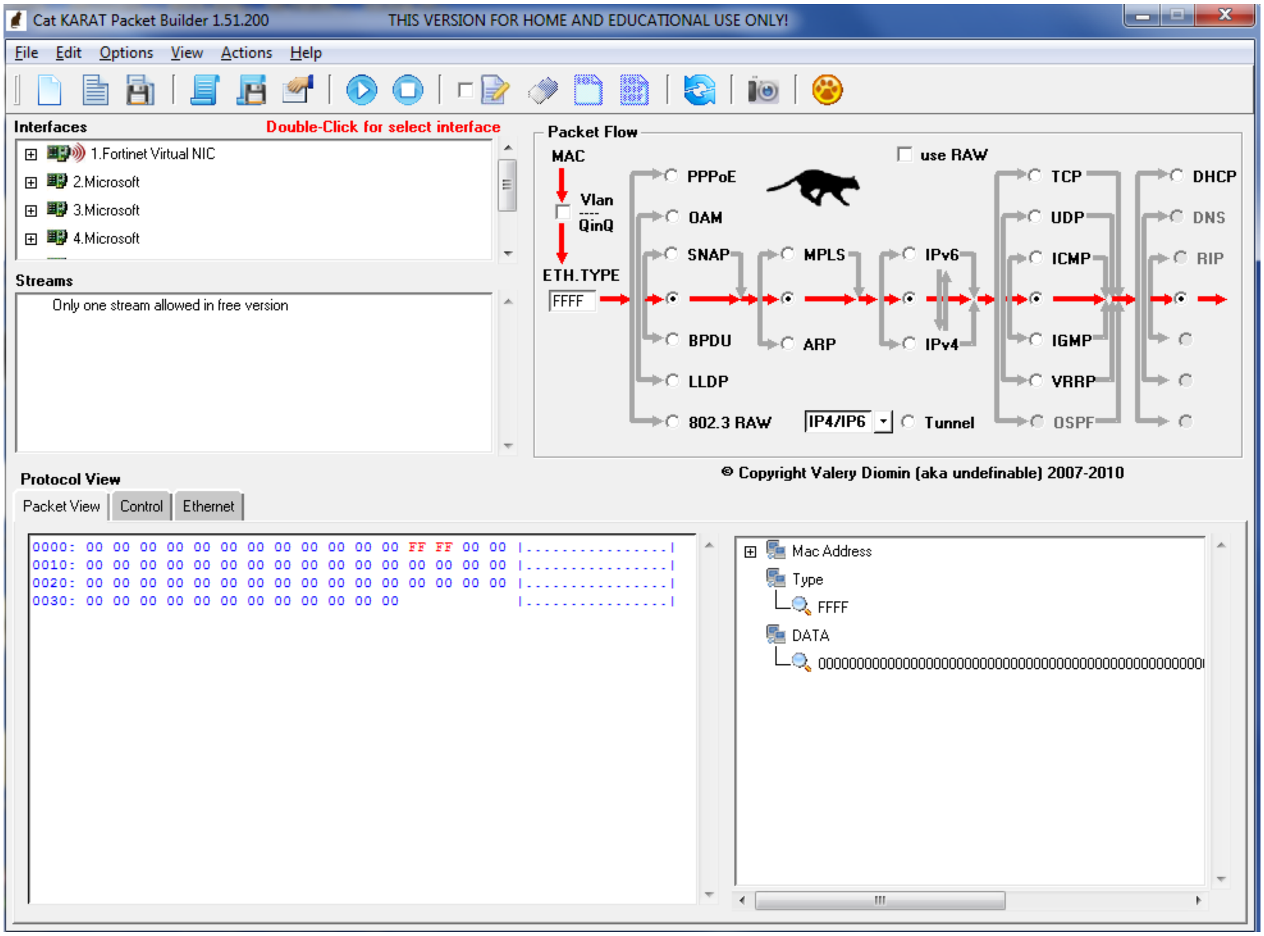
The following procedure assumes that:

- The PolarFire FPGA is programmed with the demo design programming file (.job). For more information, see Appendix 1: Programming the Device Using FlashPro Express, page 26.
- The Cat Karat and the Wireshark software are installed on the host PC.

To run the demo, perform the following steps:

1. Ensure that the RJ45 cable is connected from the host PC to the J15 connector on the board.
2. Power up the board using the **SW3** slide switch.
3. Confirm that LED 7 is glowing, indicating the Ethernet PHY link is up and running.
4. Open the Cat Karat software from the Start menu of the host PC.
   The **Cat Karat Packet Builder** window opens, as shown in Figure 18.

Figure 18 • **Cat Karat Packet Builder Window**



Only one stream allowed in free version

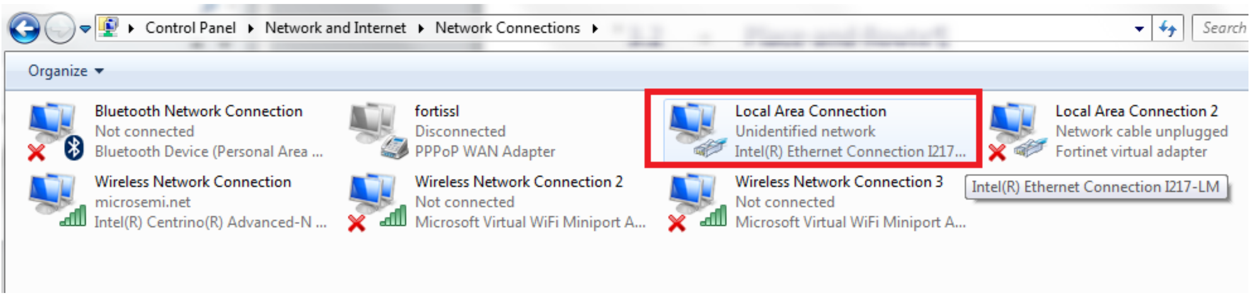
© Copyright Valery Diomin (aka undefinable) 2007-2010

```
0000:  00 00 00 00 00 00 00 00 00 00 00 00 FF FF 00 00 |................|
0010:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
0020:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
0030:  00 00 00 00 00 00 00 00 00 00 00 00             |................|
```

```
00000000000000000000000000000000000000000000000000000000
```

5. From the control panel of the host PC, note the name of the Ethernet etwork connection, as shown in Figure 19. On a Windows 10 machine, this connection is **Ethernet**.

Figure 19 • **Host PC Ethernet Network Connection**


Control Panel ▸ Network and Internet ▸ Network Connections ▸
Search
Organize ▾

Bluetooth Network Connection
Not connected
Bluetooth Device (Personal Area ...

fortissl
Disconnected
PPPoP WAN Adapter

Local Area Connection
Unidentified network
Intel(R) Ethernet Connection I217...

Local Area Connection 2
Network cable unplugged
Fortinet virtual adapter

Wireless Network Connection
microsemi.net
Intel(R) Centrino(R) Advanced-N ...

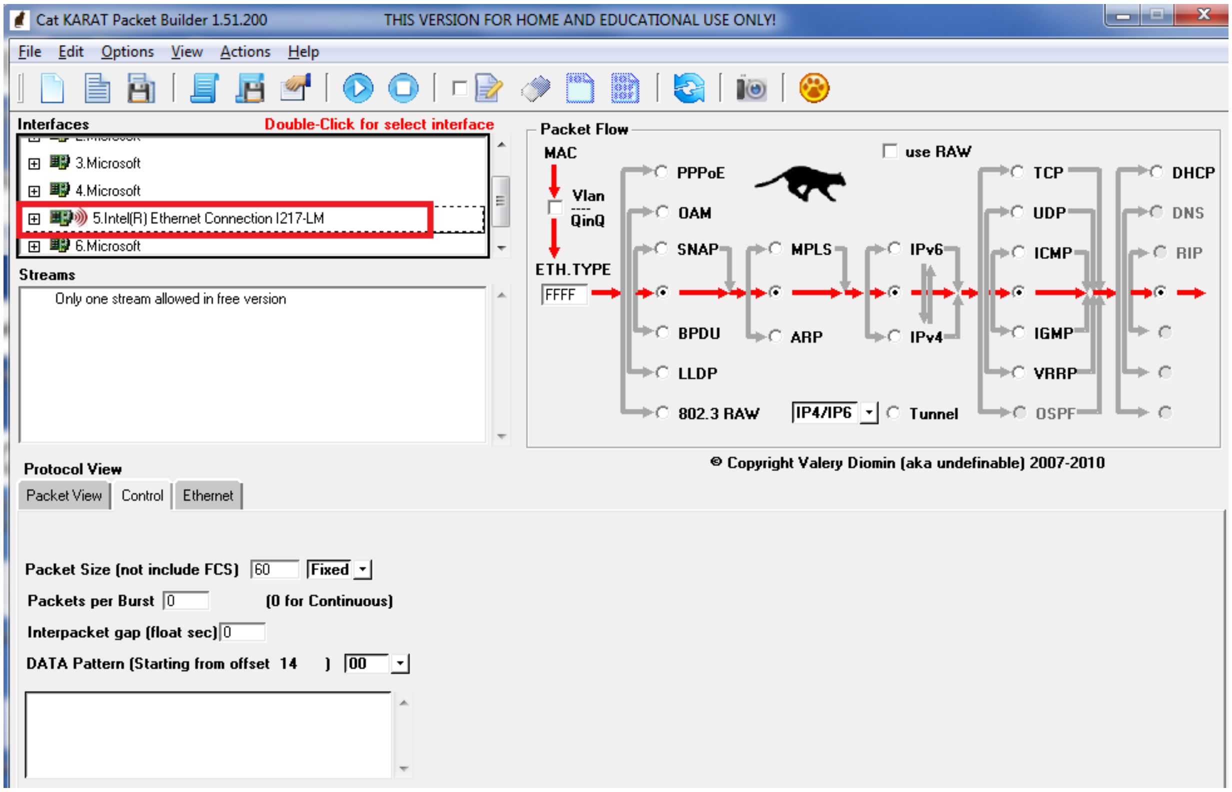
Wireless Network Connection 2
Not connected
Microsoft Virtual WiFi Miniport A...

Wireless Network Connection 3
Not connected
Microsoft Virtual WiFi Miniport A...

Intel(R) Ethernet Connection I217-LM

6. In the **Cat Karat Packet Builder** window > **Interfaces** pane, double-click **Ethernet Network connection** noted in the previous step to select that interface, as shown in Figure 20.

Figure 20 • **Interface Selection**



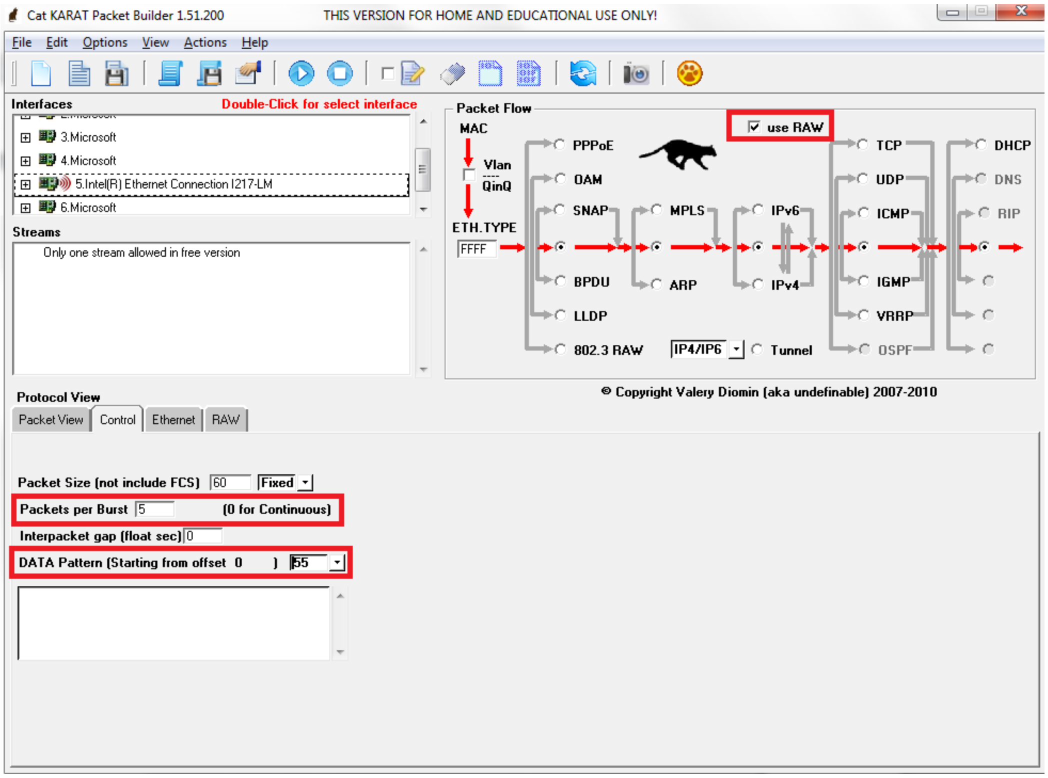
Only one stream allowed in free version

© Copyright Valery Diomin (aka undefinable) 2007-2010

Packet Size (not include FCS) 60 Fixed
Packets per Burst 0 (0 for Continuous)
Interpacket gap (float sec) 0
DATA Pattern (Starting from offset 14) 00

7. In the **Packet Flow** pane, select **use RAW** checkbox, and set **Packets per Burst** to **5** and the **Data Pattern** to **55**, as shown in Figure 21.

Figure 21 • Packet Flow and View Settings



Cat KARAT Packet Builder 1.51.200   THIS VERSION FOR HOME AND EDUCATIONAL USE ONLY!

File Edit Options View Actions Help

Interfaces   Double-Click for select interface

- 3.Microsoft
- 4.Microsoft
- 5.Intel(R) Ethernet Connection I217-LM
- 6.Microsoft

Streams

Only one stream allowed in free version

Packet Flow

☑ use RAW

© Copyright Valery Diomin (aka undefinable) 2007-2010

Protocol View

Packet View | Control | Ethernet | RAW

Packet Size (not include FCS) 60 Fixed

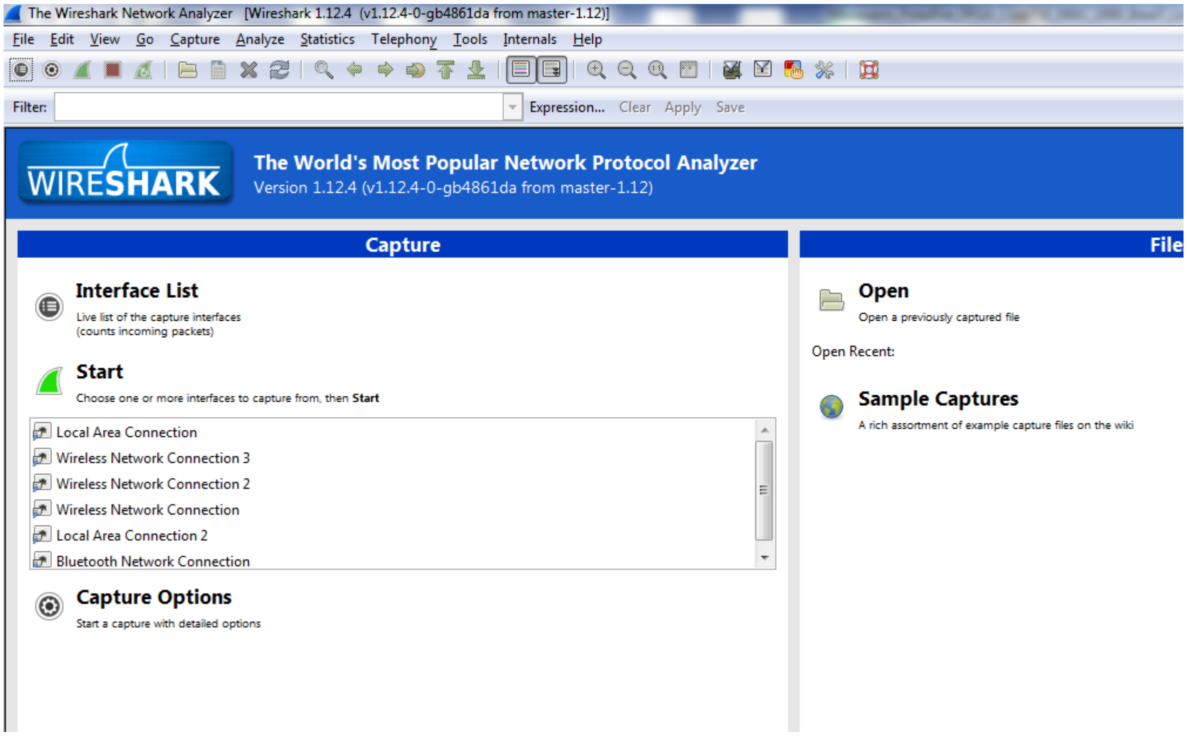
Packets per Burst 5 (0 for Continuous)

Interpacket gap (float sec) 0

DATA Pattern (Starting from offset 0) 55

8. Open the Wireshark software from the **Start** menu of the host PC. Figure 22 shows the Wireshark window.

Figure 22 • Wireshark Main Window



The Wireshark Network Analyzer [Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

WIRESHARK

The World's Most Popular Network Protocol Analyzer

Version 1.12.4 (v1.12.4-0-gb4861da from master-1.12)

Capture

Interface List

Live list of the capture interfaces (counts incoming packets)

Start

Choose one or more interfaces to capture from, then **Start**

- Local Area Connection
- Wireless Network Connection 3
- Wireless Network Connection 2
- Wireless Network Connection
- Local Area Connection 2
- Bluetooth Network Connection

Capture Options

Start a capture with detailed options

File

Open

Open a previously captured file

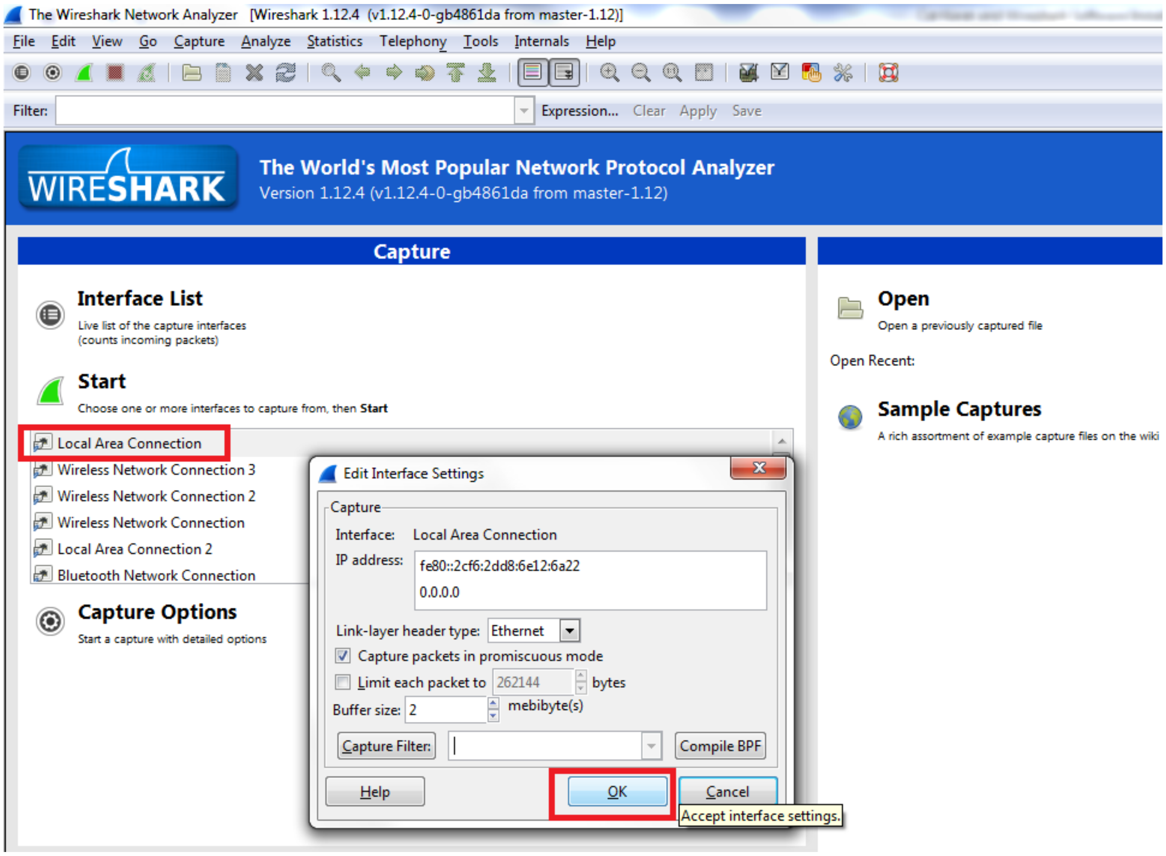
Open Recent:

Sample Captures

A rich assortment of example capture files on the wiki

9. Double-click **Local Area Connection** and select the interface settings, as shown in Figure 23. On a Window 10 machine, select **Ethernet**.

Figure 23 • **Wireshark Interface Settings**



The Wireshark Network Analyzer [Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
WIRESHARK
The World's Most Popular Network Protocol Analyzer
Version 1.12.4 (v1.12.4-0-gb4861da from master-1.12)
Capture
Interface List
Live list of the capture interfaces (counts incoming packets)
Start
Choose one or more interfaces to capture from, then Start
Local Area Connection
Wireless Network Connection 3
Wireless Network Connection 2
Wireless Network Connection
Local Area Connection 2
Bluetooth Network Connection
Capture Options
Start a capture with detailed options
Open
Open a previously captured file
Open Recent:
Sample Captures
A rich assortment of example capture files on the wiki
Edit Interface Settings
Capture
Capture Filter:
Compile BPF
Help
OK
Cancel
Accept interface settings.

Interface: Local Area Connection

IP address: fe80::2cf6:2dd8:6e12:6a22
0.0.0.0

Link-layer header type: Ethernet

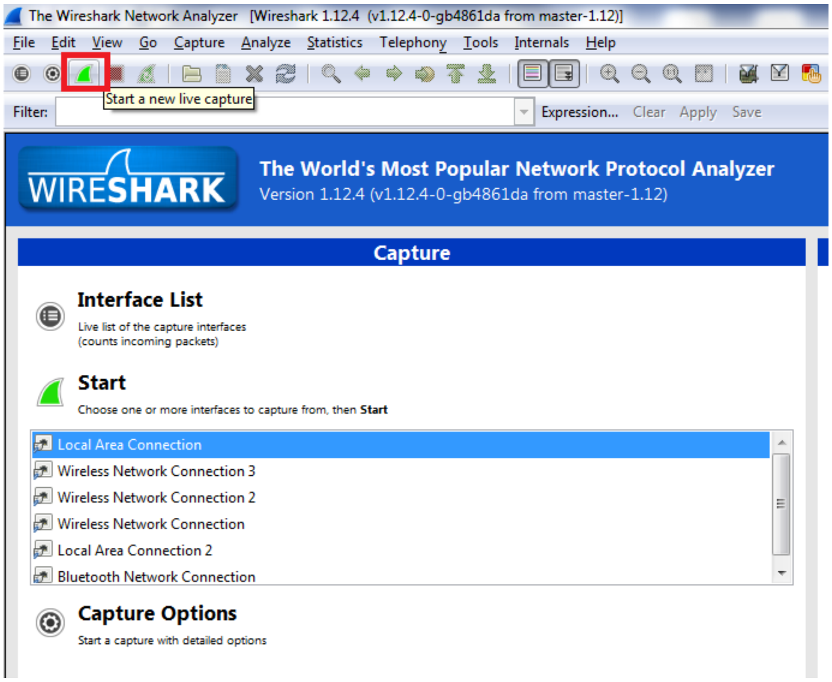
☑ Capture packets in promiscuous mode

☐ Limit each packet to 262144 bytes

Buffer size: 2 mebibyte(s)

10. Click **Start a new live capture** icon, as shown in Figure 24.

Figure 24 • **Wireshark - Start a New Live Capture Icon**

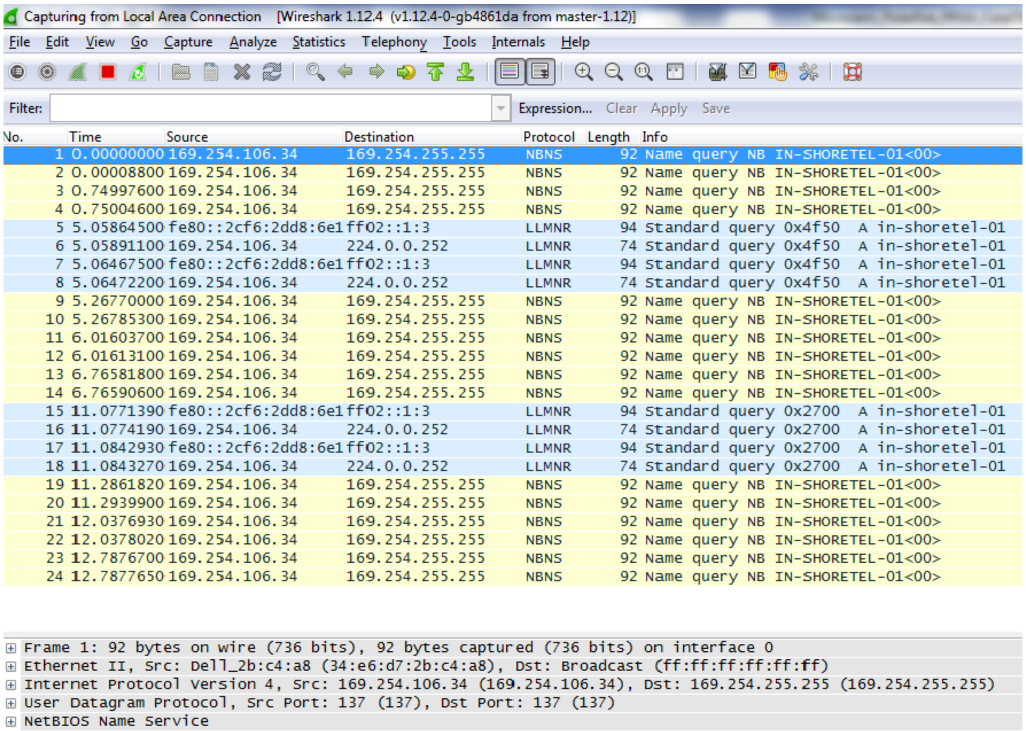


The Wireshark Network Analyzer [Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Start a new live capture
Filter:
Expression... Clear Apply Save
WIRESHARK
The World's Most Popular Network Protocol Analyzer
Version 1.12.4 (v1.12.4-0-gb4861da from master-1.12)
Capture
Interface List
Live list of the capture interfaces (counts incoming packets)
Start
Choose one or more interfaces to capture from, then Start
Local Area Connection
Wireless Network Connection 3
Wireless Network Connection 2
Wireless Network Connection
Local Area Connection 2
Bluetooth Network Connection
Capture Options
Start a capture with detailed options

The Wireshark live capture displays the Ethernet packets transferred from the board to the host PC network card, as shown in Figure 25.

Figure 25 • Wireshark Live Capture



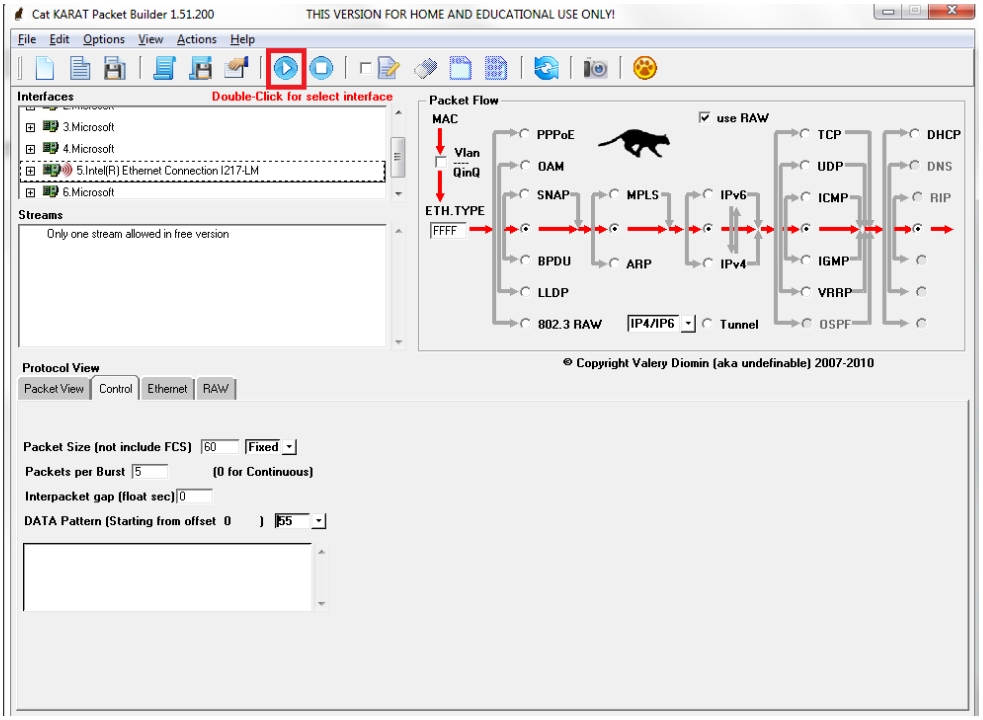
Capturing from Local Area Connection [Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.   Time       Source                Destination      Protocol Length Info
    1 0.00000000 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
    2 0.00008800 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
    3 0.74997600 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
    4 0.75004600 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
    5 5.05864500 fe80::2cf6:2dd8:6e1   ff02::1:3        LLMNR        94 Standard query 0x4f50  A in-shoretel-01
    6 5.05891100 169.254.106.34        224.0.0.252      LLMNR        74 Standard query 0x4f50  A in-shoretel-01
    7 5.06467500 fe80::2cf6:2dd8:6e1   ff02::1:3        LLMNR        94 Standard query 0x4f50  A in-shoretel-01
    8 5.06472200 169.254.106.34        224.0.0.252      LLMNR        74 Standard query 0x4f50  A in-shoretel-01
    9 5.26770000 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
   10 5.26785300 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
   11 6.01603700 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
   12 6.01613100 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
   13 6.76581800 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
   14 6.76590600 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
   15 11.0771390 fe80::2cf6:2dd8:6e1   ff02::1:3        LLMNR        94 Standard query 0x2700  A in-shoretel-01
   16 11.0774190 169.254.106.34        224.0.0.252      LLMNR        74 Standard query 0x2700  A in-shoretel-01
   17 11.0842930 fe80::2cf6:2dd8:6e1   ff02::1:3        LLMNR        94 Standard query 0x2700  A in-shoretel-01
   18 11.0843270 169.254.106.34        224.0.0.252      LLMNR        74 Standard query 0x2700  A in-shoretel-01
   19 11.2861820 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
   20 11.2939900 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
   21 12.0376930 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
   22 12.0378020 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
   23 12.7876700 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>
   24 12.7877650 169.254.106.34        169.254.255.255  NBNS         92 Name query NB IN-SHORETEL-01<00>

⊞ Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
⊞ Ethernet II, Src: Dell_2b:c4:a8 (34:e6:d7:2b:c4:a8), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol Version 4, Src: 169.254.106.34 (169.254.106.34), Dst: 169.254.255.255 (169.254.255.255)
⊞ User Datagram Protocol, Src Port: 137 (137), Dst Port: 137 (137)
⊞ NetBIOS Name Service
```

11. In the **Cat Karat** window, click **Start Transmit** to transmit five packets from the host PC to the board, as shown in Figure 26.

Figure 26 • Car Karat - Transmit Packets Icon



Cat KARAT Packet Builder 1.51.200
THIS VERSION FOR HOME AND EDUCATIONAL USE ONLY!
File Edit Options View Actions Help
Interfaces
Double-Click for select interface
⊞ 3.Microsoft
⊞ 4.Microsoft
⊞ 5.Intel(R) Ethernet Connection I217-LM
⊞ 6.Microsoft
Streams
Only one stream allowed in free version
Packet Flow
MAC
Vlan
QinQ
ETH.TYPE
FFFF
☑ use RAW
PPPoE
OAM
SNAP
BPDU
LLDP
802.3 RAW
MPLS
ARP
IP4/IP6
Tunnel
IPv6
IPv4
TCP
UDP
ICMP
IGMP
VRRP
OSPF
DHCP
DNS
RIP
© Copyright Valery Diomin (aka undefinable) 2007-2010
Protocol View
Packet View
Control
Ethernet
RAW

Packet Size (not include FCS) 60 Fixed

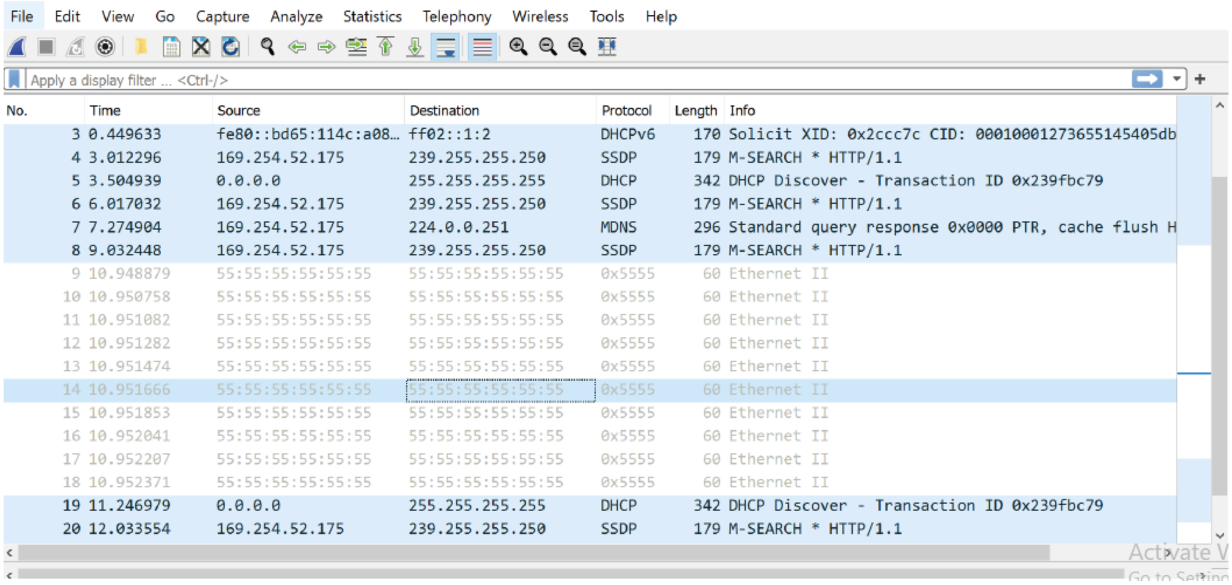
Packets per Burst 5 (0 for Continuous)

Interpacket gap (float sec) 0

DATA Pattern (Starting from offset 0) 55

12. Verify that 5 packets have been captured (transmitted and received), as shown in Figure 27.

Figure 27 • **Transmitted and Looped Back Packets**



No.	Time	Source	Destination	Protocol	Length	Info
3	0.449633	fe80::bd65:114c:a08…	ff02::1:2	DHCPv6	170	Solicit XID: 0x2ccc7c CID: 00010001273655145405db
4	3.012296	169.254.52.175	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
5	3.504939	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x239fbc79
6	6.017032	169.254.52.175	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
7	7.274904	169.254.52.175	224.0.0.251	MDNS	296	Standard query response 0x0000 PTR, cache flush H
8	9.032448	169.254.52.175	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
9	10.948879	55:55:55:55:55:55	55:55:55:55:55:55	0x5555	60	Ethernet II
10	10.950758	55:55:55:55:55:55	55:55:55:55:55:55	0x5555	60	Ethernet II
11	10.951082	55:55:55:55:55:55	55:55:55:55:55:55	0x5555	60	Ethernet II
12	10.951282	55:55:55:55:55:55	55:55:55:55:55:55	0x5555	60	Ethernet II
13	10.951474	55:55:55:55:55:55	55:55:55:55:55:55	0x5555	60	Ethernet II
14	10.951666	55:55:55:55:55:55	55:55:55:55:55:55	0x5555	60	Ethernet II
15	10.951853	55:55:55:55:55:55	55:55:55:55:55:55	0x5555	60	Ethernet II
16	10.952041	55:55:55:55:55:55	55:55:55:55:55:55	0x5555	60	Ethernet II
17	10.952207	55:55:55:55:55:55	55:55:55:55:55:55	0x5555	60	Ethernet II
18	10.952371	55:55:55:55:55:55	55:55:55:55:55:55	0x5555	60	Ethernet II
19	11.246979	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x239fbc79
20	12.033554	169.254.52.175	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1

The preceding figure highlights five packets that were transmitted from the host PC to the board, looped back at the CoreTSE IP, and sent back to the host PC. All packets transmitted from the host PC network are looped back in the same way.

13. Select different burst rates, data patterns, and transmit packets to the board.
14. Power down the board and close the Cat Karat and the Wireshark software.

You have successfully run the demo.

5 Appendix 1: Programming the Device Using FlashPro Express

This chapter describes how to program the PolarFire device with the job file using Flashpro Express. The job file is available at the following design files folder location:

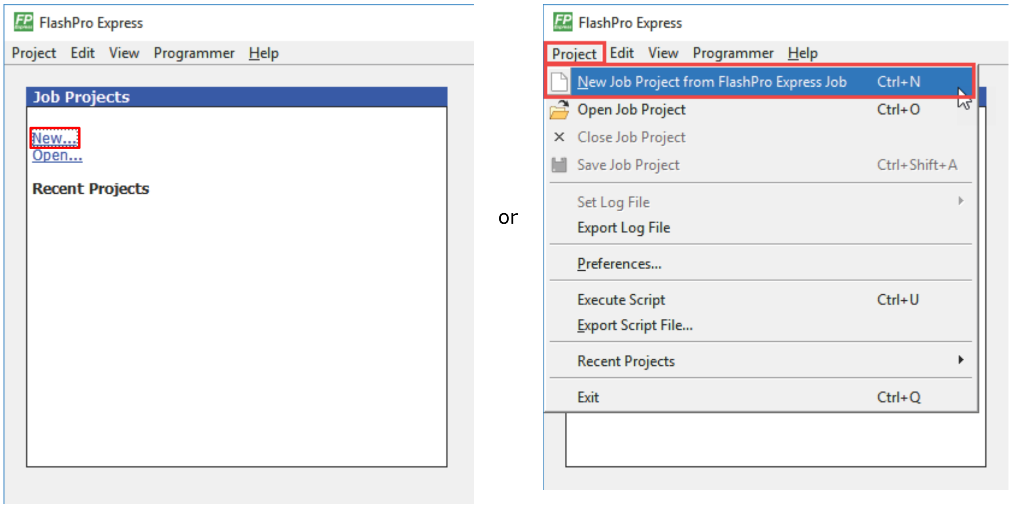
`mpf_dg0799_eval_df\Programming_Job`

Follow these steps:

1. Connect the jumpers and set up the PolarFire Evaluation Kit Board as described in steps 1 to 5 of Run PROGRAM Action, page 18.
2. On the host PC, start the FlashPro Express software from `<$Installation Directory>\Microsemi\Designer\binfp`.
3. To create a new job project, select **New** or

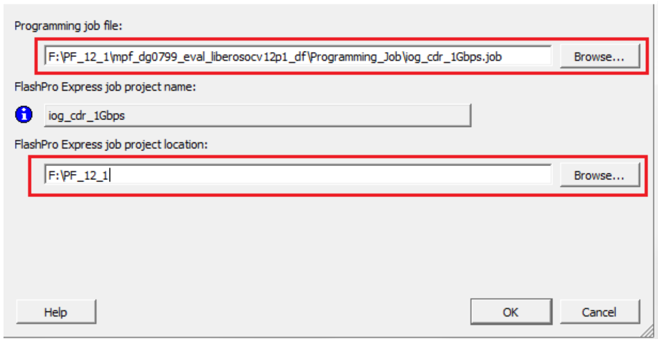
   In the **Project** menu, select **New Job Project from FlashPro Express Job**, as shown in Figure 28.

Figure 28 • **FlashPro Express Job Project**



4. 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_dg0799_eval_df\Programming_Job`
   - **FlashPro Express job project location**: Select **Browse** and navigate to the location where you want to save the project.

Figure 29 • **New Job Project from FlashPro Express Job**


Programming job file: F:\PF_12_1\mpf_dg0799_eval_liberosocv12p1_df\Programming_Job\iog_cdr_1Gbps.job

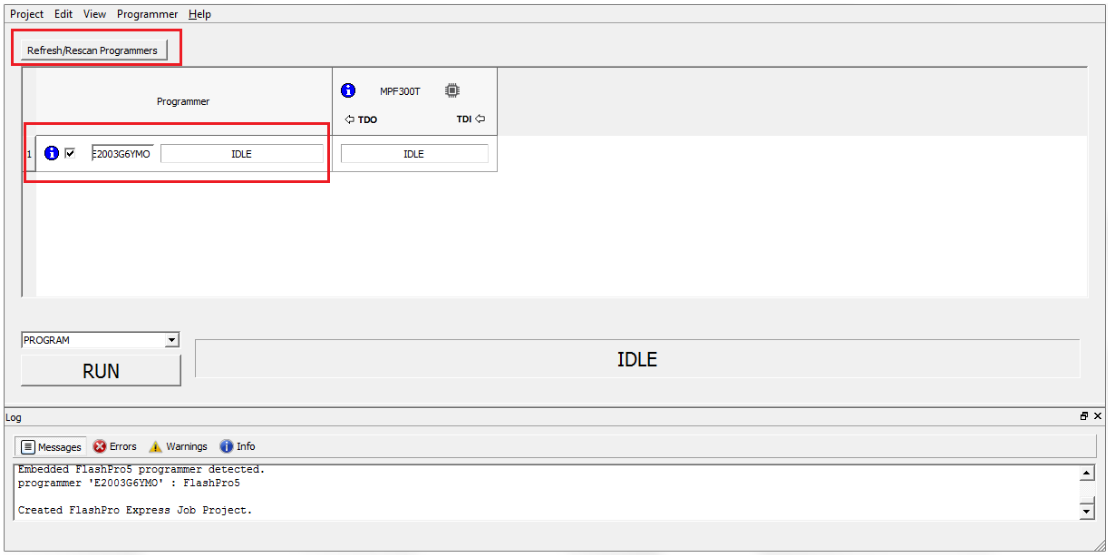
FlashPro Express job project name: iog_cdr_1Gbps

FlashPro Express job project location: F:\PF_12_1

5. Click **OK**. The required programming file is selected and ready to be programmed in the device.

6. The FlashPro Express window appears as shown in Figure 30. 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**

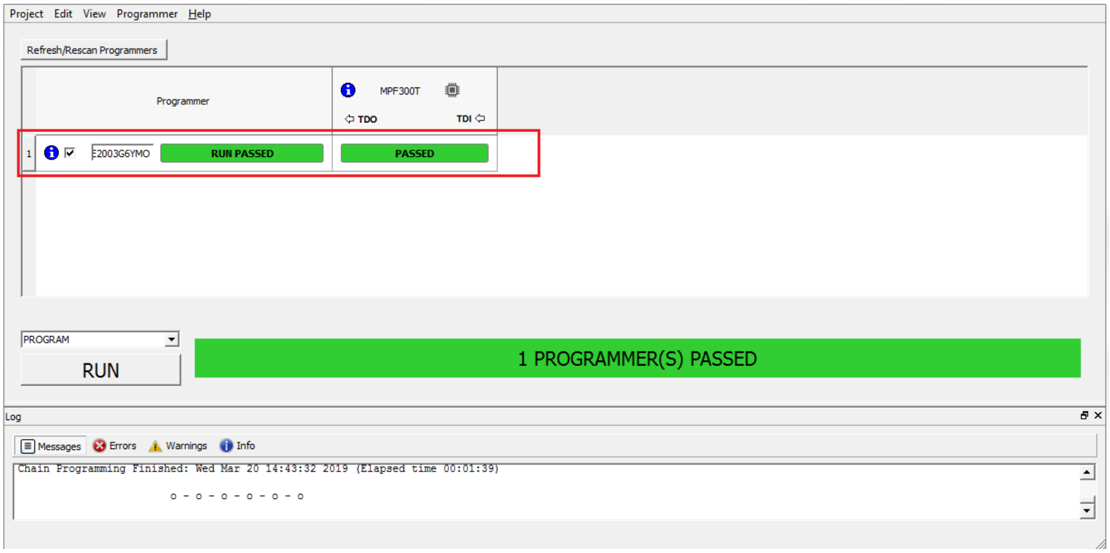


```
Embedded FlashPro5 programmer detected.
programmer 'E2003G6YMO' : FlashPro5

Created FlashPro Express Job Project.
```

7. Click **RUN.** When the device is programmed successfully, a **RUN PASSED** status is displayed as shown in Figure 31. See Running the Demo, page 20.

Figure 31 • **FlashPro Express—RUN PASSED**



```
Chain Programming Finished: Wed Mar 20 14:43:32 2019 (Elapsed time 00:01:39)
                      o - o - o - o - o - o
```

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

6 Appendix 2: Running the TCL Script

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

To run the TCL, follow the steps below:

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

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

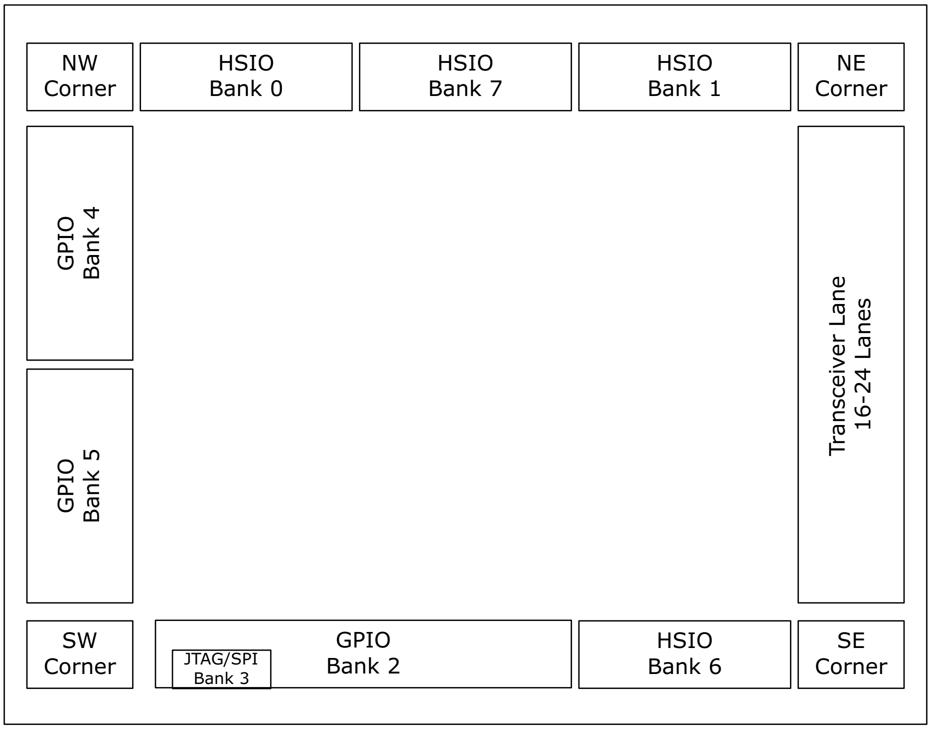
For more information about TCL scripts, refer to **mpf_dg0799_df/TCL_Scripts/readme.txt.**

Refer to *Libero® SoC TCL Command Reference Guide* for more details on TCL commands. Contact Technical Support for any queries encountered when running the TCL script.

7 Appendix 3: Multi-Lane 1G IOD CDR Design

In a multi-lane design, Ethernet traffic from multiple RJ45 cables comes into the FPGA through PHY. In such cases, multiple RX and TX ports must be assigned from the PolarFire GPIO Banks to form multiple SGMII links with the PHY. The following figure shows the placement of I/O Banks and PLLs in a PolarFire device (MPF300).

Figure 32 • **I/O Banks and PLL placement in MPF300**



Each Bank has multiple I/O Lanes and each I/O Lane includes 6 I/O pairs. The lane controller available in each I/O lane has a clock recovery unit, which is used for the Clock recovery of that lane. Hence, only one SGMII link can be realized from an I/O lane.

For an 8-lane design, 8 I/O Lanes are used to form 8 SGMII links. To enable sharing of the PF_IOD_CDR_CCC, the selection of these I/O Lanes must be made in any of the following ways:

- Lanes of the same Bank can be selected vertically up to half of the side
- Lanes of the same Bank can be selected horizontally up to half of the side
- Lanes from vertical and horizontal Banks can be selected

Note: In Libero SoC, when I/O lanes are selected from Bank 5 or 2, or from both and placed, the PF_IOD_CDR_CCC selects the SW PLL and is placed SW. If all I/O lanes are selected from Bank 4, PF_IOD_CDR_CCC selects the NW PLL and is placed NW.

When the reference clock for all the links is the same:

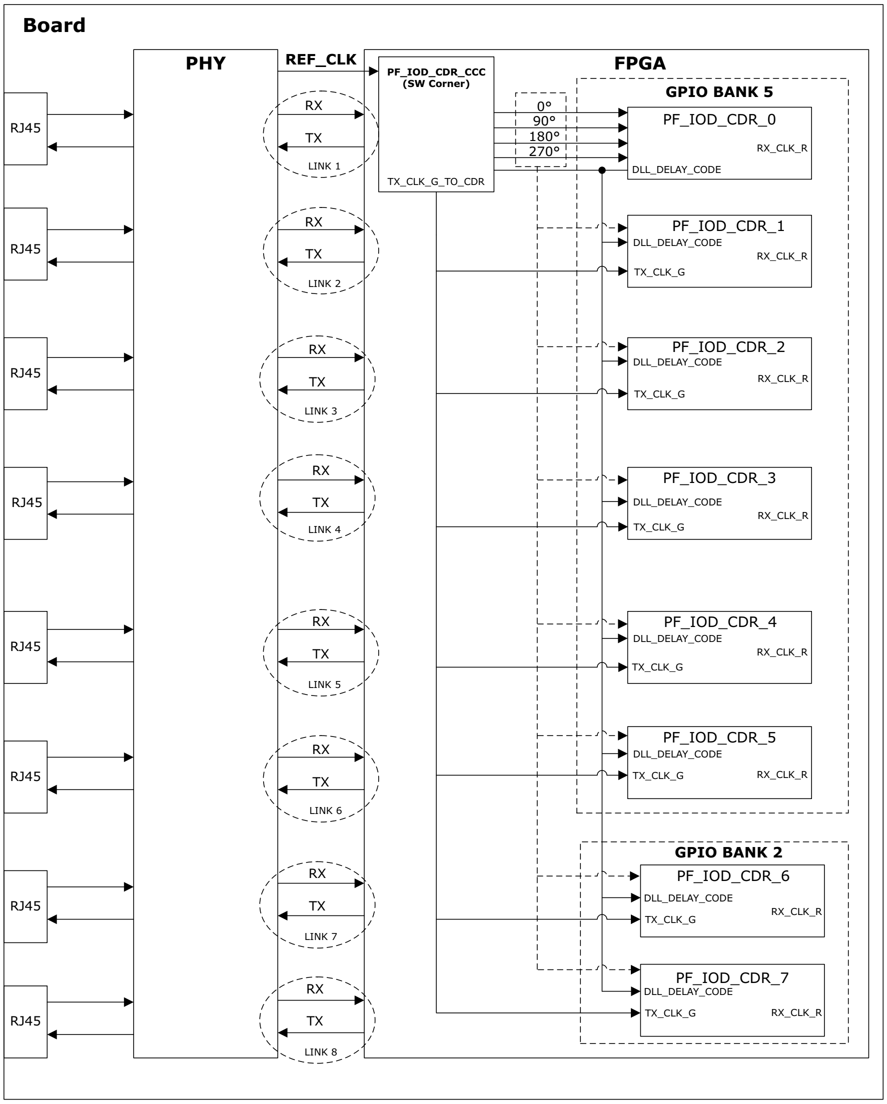
- PF_IOD_CDR_CCC can be shared across all IOD blocks (PF_IOD_CDR) for the HSIO BANK clocks and transmit clock (TX_CLK).
- PF_IOD_CDR_CCC uses an internal lane controller to generate the DLL delay code and share it with all IOD blocks. The DLL delay code is required for phase tuning/adjustment.

To conclude, the following IOD resources are used to create an 8-lane design in a PolarFire MPF300 device:

- One PF_IOD_CDR_CCC with a lane controller for DLL delay update
- 8 I/O Lanes and lane controllers for clock recovery

The following figure shows the high-level block diagram of an 8-lane design implemented using Libero SoC PolarFire.

Figure 33 • **8 Lane 1G IOD CDR Design in PolarFire**



As shown in Figure 33, eight PF_IOD_CDR instances are instantiated from GPIO Banks 5 and 2 to form eight links. The Clock Conditioning Circuit (CCC) available in the South-West corner, is configured in the PLL-DLL cascaded mode for the clock recovery and DLL delay update.

Apart from 8 lane controllers for clock recovery, an additional lane controller from the PF_IOD_CDR_CCC is inferred during synthesis for sharing the DLL delay update. This optimizes the utilization of lane controllers in the device.

8 Appendix 4: 1G Ethernet BASE-T and BASE-X Using Transceiver

The PolarFire FPGA family includes multiple embedded low-power, performance-optimized transceivers. Each transceiver has both the Physical Medium Attachment (PMA), protocol Physical Coding Sub-layer (PCS) logic, and interfaces to the FPGA fabric. The transceiver has a multi-lane architecture with each lane natively supporting serial data transmission rates from 250 Mbps to 12.7 Gbps. For more information, see *UG0677: PolarFire FPGA Transceiver User Guide*.

This section describes how 1G Ethernet BASE-T and BASE-X designs are implemented in PolarFire FPGAs using the transceivers.

8.1 1G Ethernet BASE-T and BASE-X

Figure 34 shows the typical FPGA design for 1G Ethernet BASE-T.

Figure 34 • **1G BASE-T Design**

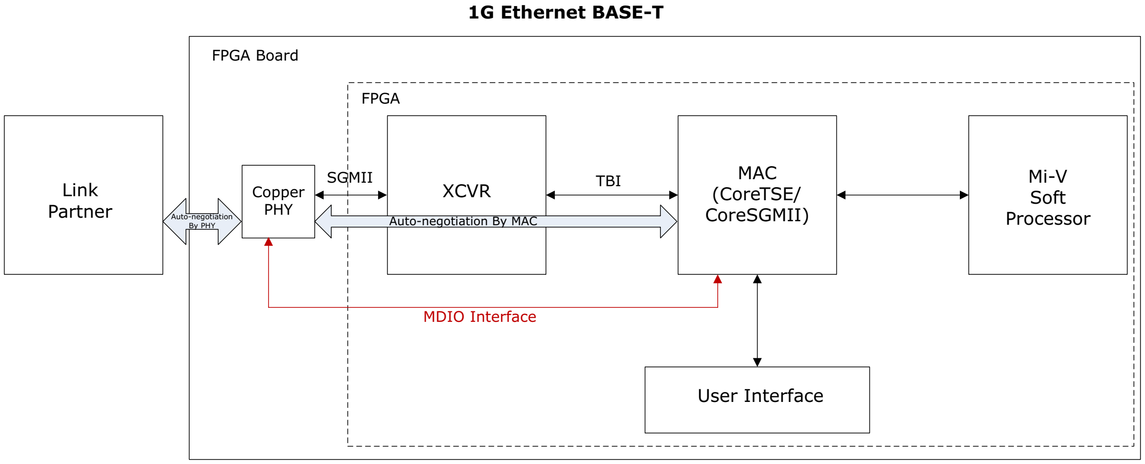
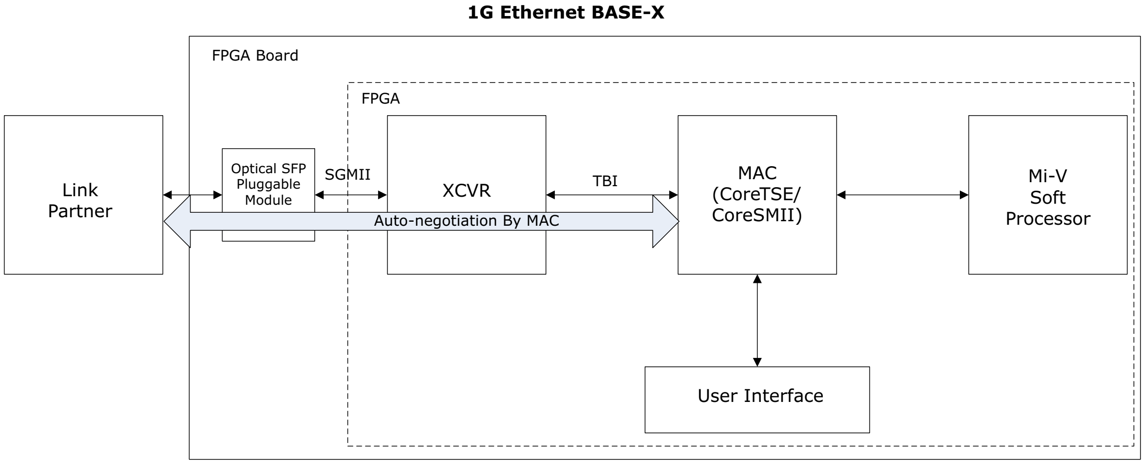


Figure 35 shows the typical FPGA design for 1G Ethernet BASE-X.

Figure 35 • **1G BASE-X Design**



The following points summarize the 1G Ethernet BASE-T and BASE-X designs:

- The Mi-V soft processor is used to configure the PHY registers (using MDIO interface), MAC Configuration and Management registers. Users can also implement a fabric logic or any other soft processor to implement these functions.
- The MAC IP is configured in the Ten Bit Interface mode (TBI).
- In BASE-T, the management block of the MAC IP auto-negotiates with the onboard PHY as per Clause 28 of the IEEE802.3z standard. The PHY auto-negotiates with the link partner.
- In BASE-X, the management block of the MAC IP auto-negotiates with the link partner as per Clause 37 of the IEEE802.3z standard.
- The auto-negotiation (AN) functions are defined in the MAC Management registers 04h to 08h. The bit field formats of these registers are different for BASE-T and BASE-X. For more information about the bit field formats of the AN registers, see *HB0549: CoreTSE v3.1 Handbook* or *HB0627: CoreSGMII v3.2 Handbook.* The following table lists the AN registers.

  Table 5 • **AN Registers**

Register	Description
04h	AN Advertisement
05h	AN Link Partner Base Page Ability
06h	AN Expansion
07h	AN Next Page Transmit
08h	AN Link Partner Ability Next Page

  The following registers are common in BASE-T and BASE-X modes:

  - Control register at address 0x00
  - Status register at address 0x01

  Registers 0x04, 0x05, 0x06, 0x07, and 0x08 are based on the configuration.
- XCVR is configured to operate at 1250 Mbps. For more information about XCVR configuration, see Transceiver Configuration, page 33.
- The user data from MAC (CoreTSE non-AHB) is provided on a 32-bit parallel bus.

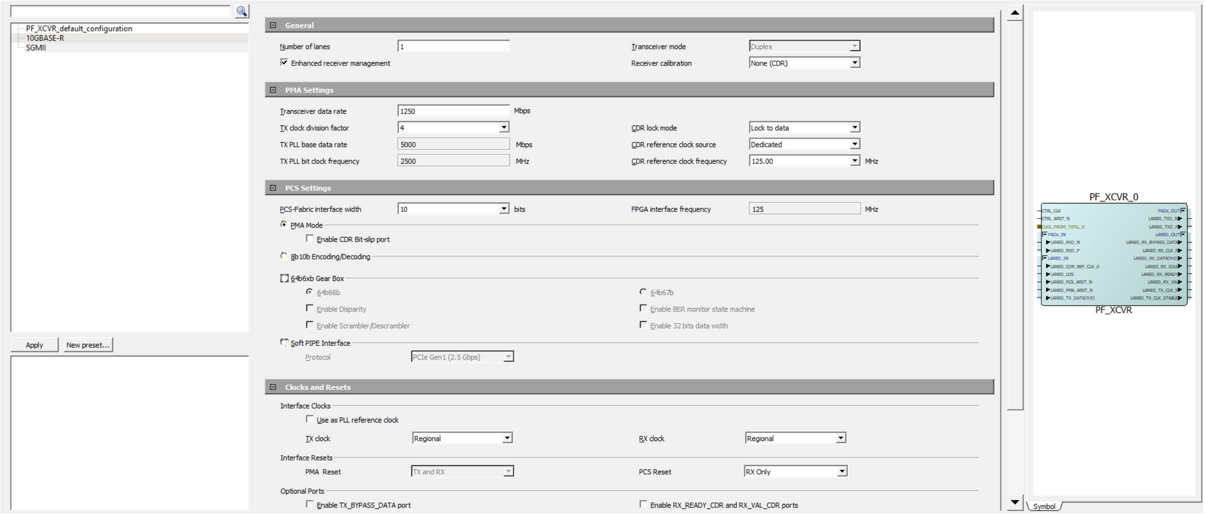
8.2 Transceiver Configuration

For 1G Ethernet BASE-T and BASE-X, the transceiver block is configured for 1.25 Gbps data rate. Figure 36 shows the configuration of the transceiver block in the Libero SoC design suite. The following table lists the transceiver configuration.

Table 6 • **XCVR Configuration**

Parameters	Settings
Number of lanes	1
Data Rate	1250 Mbps
PMA	
TX clock division factor	4
TX PLL base data rate	5000 Mbps
TX PLL bit clock frequency	2500 Mbps
CDR lock mode	lock to data
CDR reference clock source	Dedicated
CDR reference clock frequency	125 MHz
PCS	
PCS-fabric interface width	10 bits
FPGA interface frequency	125 MHz
PMA Mode	Enabled
Clocks and Resets	
TX clock	Regional
RX clock	Regional
PCS Reset	RX Only

Figure 36 • **Transceiver Configuration**



PF_XCVR_default_configuration
10GBASE-R
SGMII

General

Number of lanes: 1
☑ Enhanced receiver management
Transceiver mode: Duplex
Receiver calibration: None (CDR)

PMA Settings

Transceiver data rate: 1250 Mbps
TX clock division factor: 4
TX PLL base data rate: 5000 Mbps
TX PLL bit clock frequency: 2500 MHz
CDR lock mode: Lock to data
CDR reference clock source: Dedicated
CDR reference clock frequency: 125.00 MHz

PCS Settings

PCS-Fabric interface width: 10 bits
FPGA interface frequency: 125 MHz
● PMA Mode
☐ Enable CDR Bit-slip port
○ 8b10b Encoding/Decoding
○ 64b6xb Gear Box
● 64b66b
○ 64b67b
☐ Enable Disparity
☐ Enable BER monitor state machine
☐ Enable Scrambler/Descrambler
☐ Enable 32 bits data width
○ Soft PIPE Interface
Protocol: PCIe Gen1 (2.5 Gbps)

Clocks and Resets

Interface Clocks
☐ Use as PLL reference clock
TX clock: Regional
RX clock: Regional
Interface Resets
PMA Reset: TX and RX
PCS Reset: RX Only
Optional Ports
☐ Enable TX_BYPASS_DATA port
☐ Enable RX_READY_CDR and RX_VAL_CDR ports

8.3 Transceiver Connections

This section describes the typical transceiver to CoreTSE connections in BASE-T and BASE-X design.

The following table lists the transceiver input and output port connections.

Table 7 • **XCVR Port Connections**

Port Name	Input or Output	Connection Description
CTRL_CLK	Input	40 MHz clock for the enhanced receiver management logic Can be sourced from the on-chip 160 MHz RC oscillator through a clock divider. or It can be connected to the output fabric clock of CCC.
CTRL_ARST_N		Input signal to reset ERM. Drive this signal from the XCVR_INIT_DONE signal of the PF_INIT_MONITOR component.
CLKS_FROM_TX_PLL		XCVR transmit clock sourced from the TX PLL
LANE0_RXD_N LANE0_RXD_P		Differential receive input pads for receiving the Ethernet data
LANE0_CDR_REF_CLK		125 MHz reference for clock and data recovery
LANE0_PCS_ARST_N		Asynchronous active-low reset signal used to reset the PCS module of XCVR lane
LANE0_PMA_ARST_N		Asynchronous active-low reset signal used to reset the PMA module of XCVR lane
LANE0_RX_DATA [9:0]		The 10-bit RX data from XCVR to CoreTSE:RCG [9:0]
LANE0_TXD_N LANE0_TXD_P		Differential transmit output pads
LANE0_RX_CLK_R		Recovered regional receive clock from XCVR to the fabric logic and CoreTSE:TBI_RX_CLK
LANE0_TX_DATA [9:0]		The 10-bit TX data from CoreTSE:TCG [9:0] to XCVR

For other XCVR ports, see *UG0677: PolarFire FPGA Transceiver User Guide*.

9 Appendix 5: References

This section lists the documents that provide more information about the IP cores used in the 1G loopback demo design and about PolarFire 1G Ethernet Solutions in general.

- For more information about PF_IOD_CDR_CCC and PF_IOD_CDR, see *UG0686: PolarFire FPGA User I/O User Guide*.
- For more information about CoreTSE, see *HB0549: CoreTSE Handbook*.
- For more information about PF_CCC, see *UG0684: PolarFire FPGA Clocking Resources User Guide*.
- For more information about PF_SRAM_AXI_AHBL, see *UG0680: PolarFire FPGA Fabric User Guide*.
- For more information about CoreAHBLite, see *CoreAHBLite Handbook*.
- For information about COREAHBTOAPB3, see *COREAHBTOAPB3 Handbook*.
- For more information about CoreAPB3, see *CoreAPB3 Handbook*.
- For more information about CoreUARTapb, see *CoreUARTapb Handbook*.
- For more information about CoreSPI, see *HB0089: CoreSPI Handbook*.
- For more information about PF_INIT_MONITOR, see *UG0725: PolarFire FPGA Device Power-Up and Resets User Guide*.
- For more information about MIV_RV32, see *MIV_RV32 Handbook* from the Libero SoC Catalog.
- For general information about PolarFire 1G Ethernet Solutions, see *UG0687: PolarFire FPGA 1G Ethernet Solutions User Guide*.
- For more information about the PolarFire Evaluation board, see *UG0747: PolarFire FPGA Evaluation Kit User Guide*.