Running Webserver and IAP Using TFTP on PolarFire Device

AN4569

Introduction (Ask a Question)

Microchip PolarFire® FPGAs support 1G Ethernet solutions for various networking applications. In PolarFire devices, 10/100/1000 Mbps (1G) Ethernet is implemented using the CoreTSE_AHB Media Access Control (MAC) soft IP core. The CoreTSE_AHB IP implements a Serial Gigabit Media-Independent Interface (SGMII or GMII) with an Ethernet PHY. This Ethernet interface can be implemented in the FPGA by using either a transceiver (PF_XCVR IP) or a GPIO with clock and data recovery (PF_IOD_CDR IP) capability. In this demo, the 1G Ethernet solution is implemented in the FPGA design by using GPIOs with CDR capability and CoreTSE_AHB IP.

The CoreTSE_AHB IP core enables system designers to implement a broad range of Ethernet designs, from low cost 10/100 Ethernet to higher performance 1 Gigabit ports. The CoreTSE_AHB IP core is suitable for use in networking equipment such as switches, routers, and data acquisition systems.

The CoreTSE_AHB IP has the following major interfaces:

- 10/100/1000 Mbps Ethernet MAC with a Gigabit Media Independent Interface (GMII) and Ten Bit Interface (TBI) to support Serial Gigabit Media Independent Interface (SGMII), 1000BASE-T, and 1000BASE-X.
- GMII or TBI physical layer interface that connects to the Ethernet PHY
- MAC data path interface

The CoreTSE_AHB IP core is available in two different versions:

- CoreTSE_AHB: Uses AHB interface for both the transmit and receive paths.
- CoreTSE_AHB (Non-AMBA): Uses direct access to MAC with a streaming packet interface.

For more information about CoreTSE_AHB IP, see the CoreTSE AHB Handbook.

CoreTSE_AHB IP core requires license for using in Libero® SoC design. For license request, contact Microchip Technical Support.

This demo design implements a webserver application and a Trivial File Transfer Protocol (TFTP) server using the PolarFire Evaluation Kit board. For more information about this board, see UG0747: PolarFire FPGA Evaluation Kit User Guide.

This demo design demonstrates the following:

- Use of Ethernet MAC connected to a SGMII PHY.
- Integration of the CoreTSE_AHB MAC driver with the lwIP TCP/IP stack and FreeRTOS operating system.
- Implementation of webserver on the PolarFire Evaluation board.
- Implementation of the TFTP server on the PolarFire Evaluation board.
- Procedure to run webserver and the TFTP server designs on the PolarFire Evaluation board.

This 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 10. Appendix 3: Programming the Device and External SPI FlashPro Express.
- Using Libero SoC: To program the device using Libero SoC, see 6. Libero Design Flow.

Table of Contents

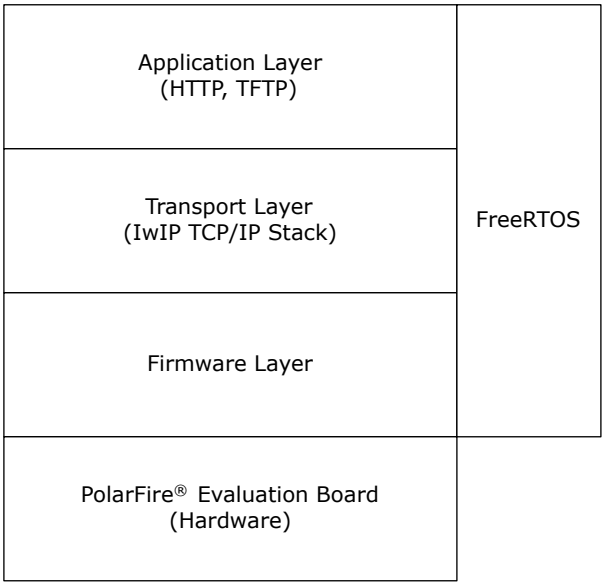
1. Demo Design Layers (Ask a Question)

The webserver and TFTP server demo design have the following layers:

- Application layer
- Transport layer (lwIP TCP/IP stack)
- Real Time Operating System (RTOS) and firmware layer

The following block diagram shows the three layers in the webserver and TFTP server applications in a PolarFire device.

Figure 1-1. Webserver and TFTP Server Applications on a PolarFire Device



1.1 Application Layer (Ask a Question)

The webserver handles the HTTP request from the client (host PC) browser and transfers the static pages to the client in response to its request. When the IP address (for example, 10.60.3.25) is typed in the address bar of a browser, an HTTP request is sent to the port associated with the webserver. The webserver interprets the request and responds to the client with the requested page or resource.

The TFTP client (the host PC) transfer files to the PolarFire device (the TFTP server) using the `TFTP PUT` command. Transferred files are stored in the PolarFire Evaluation board's external Flash memory, which is connected to the System Controller SPI interface.

1.2 Transport Layer (lwIP TCP/IP Stack) (Ask a Question)

The lwIP TCP/IP stack, developed by Adam Dunkels at the Swedish Institute of Computer Science (SICS), is suitable for embedded systems because of its low system resource usage. The lwIP stack can be used with or without an operating system. It consists of actual implementations of IP, ICMP, UDP, and TCP protocols. It supports functions such as the buffer and memory management.

lwIP is available (under a BSD license) in C source-code format for download at Index of /releases/lwip/.

1.3 RTOS and Firmware Layer (Ask a Question)

FreeRTOS is an open-source, real-time operating system kernel. In this demo, FreeRTOS is only used to prioritize and schedule tasks. For more information about FreeRTOS and the latest source code, see FreeRTOS.

The firmware provides software drivers to configure and control the following components:

- Ethernet MAC
- Core UART advanced peripheral bus (APB)
- SPI

2. Demo Design (Ask a Question)

The following is the data flow of the demo design:

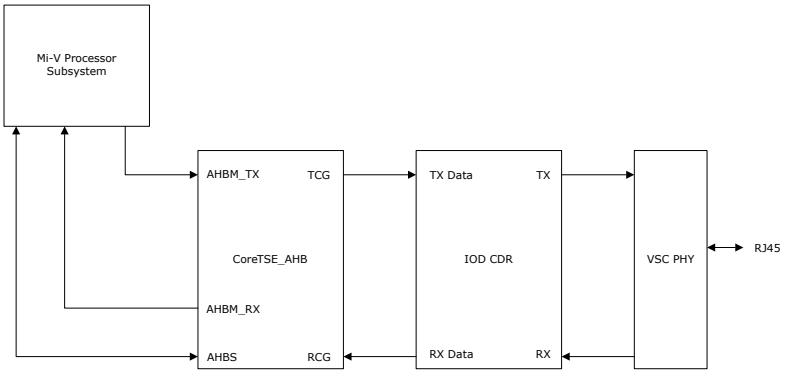
1. PF_CCC_0 provides the clock to the Mi-V processor and other APB peripherals.
2. PF_IOD_CDR_CCC drives the IOD CDR clocks PF_IOD_CDR: TX_CLK_G and HS_IO_CLK.
3. Mi-V performs the following functions:
   - Executes the application from LSRAM (PF_SRAM IP)
   - Configures the ZL30364 clock generation hardware through CoreSPI IP to generate reference clocks for the VSC PHY and the IOD CDR fabric module.
   - Configures the CoreTSE_AHB IP MAC in TBI mode and initializes the MAC in 1000 Base-T.
   - Sends a request to the CoreTSE_AHB IP to negotiate with the on-board VSC8575 PHY.
4. CoreTSE_AHB IP implements the 1G Ethernet MAC and is configured to interface with the PF_IOD_CDR block in the SGMII mode. The CoreTSE IP has an inbuilt Management Data Input/Output (MDIO) interface to exchange control and status information with the VSC PHY.
5. PF_IOD_CDR IP performs the following functions:
   - Interfaces with the on-board VSC8575 PHY.
   - Recovers the data and clock from the incoming RX_P and RX_N ports. Deserializes the recovered data and sends 10-bit parallel data to the CoreTSE.
   - Receives Ethernet data from the VSC PHY through the RX_P and RX_N input pads, gears down the receive data rate, and deserializes the data.
6. The deserialized data is sent from SGMII_CDR_0:RX_DATA[9:0] to CoreTSE_AHB IP: RCG[9:0]. The CoreTSE_AHB IP MAC receives the Ethernet packet from the on-board Ethernet PHY through high-speed PF_IOD_CDR IP using the built-in DMA controller and Mi-V processes the Ethernet packets.
7. The Ethernet packets from the Mi-V processor are sent to CoreTSE_AHB IP, and CoreTSE_AHB IP:TCG[9:0] is sent to SGMII_CDR_0:TX_DATA[9:0].
8. PF_IOD_CDR 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.

The following are the demo design features:

- Webserver
- In-application programming (IAP) using TFTP server

The following figure shows the high-level demo design block diagram. In this demo design, CoreTSE_AHB IP is instantiated in the FPGA fabric and connected to the on-board VSC PHY using IOD CDR IP.

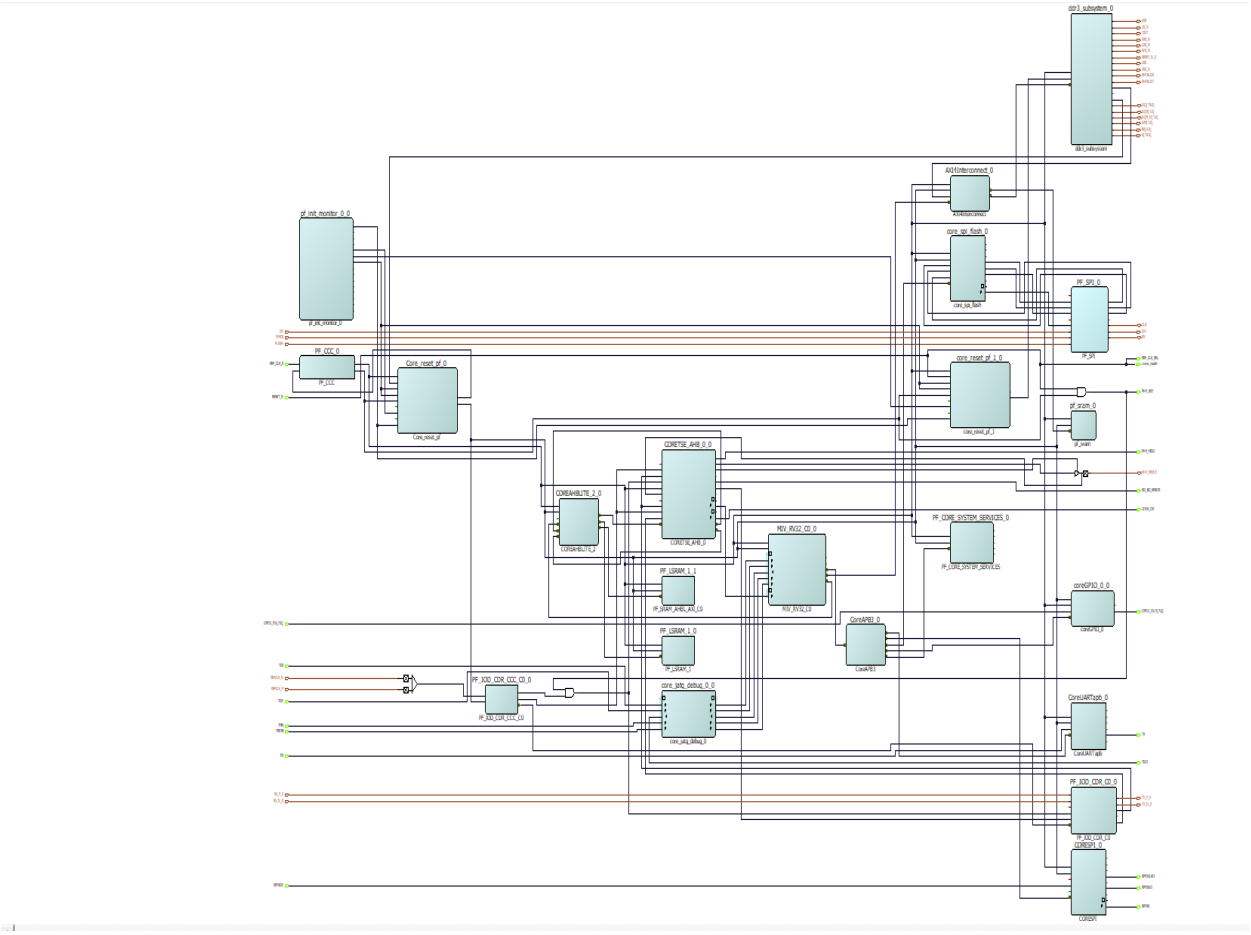
Figure 2-1. Demo Design High-level Block Diagram



2.1 Design Implementation (Ask a Question)

The following figure shows the top-level Libero implementation of the demo design. The libero project implementation is the same for both webserver and IAP using TFTP but the application firmware is different.

Figure 2-2. Top-Level Libero Implementation



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

Table 2-1. I/O Signals

Signal	Direction	Description
RX_P, RX_N	Input	IOD CDR receive signals connected to the VSC PHY transmit data signals.
REFCLK_N, REFCLK_P	Input	The 125 MHz input clock received from the on-board ZL30364 and fed to NWC_PLL_0.
RESET_N	Input	Mi-V Reset. Asserted by pressing the on-board K22 push button.
REF_CLK_0	Input	The 50 MHz input clock received from the on-board 50 MHz oscillator and fed to PF_CCC_0.
TCK, TDI, TMS, and TRSTB	Input	JTAG signals interfacing the soft processor for debugging.
TDO	Output	
TX_P, TX_N	Output	IOD CDR transmit signals connected to the VSC PHY receive data signals.
LINK_OK	Output	Link status indicator. Provides the link up or down status with the on-board PHY. This signal is mapped to the on-board LED7. The LED ON state indicates that the link is up.
PHY_RST	Output	Reset signal to the on-board VSC8575 PHY.
PHY_MDC	Output	The MDIO clock fed to the on-board VSC8575 PHY.
PHY_MDIO	Output	Management Data I/O interface for accessing the on-board VSC8575 PHY registers.
coma_mode	Output	Signal held Low (connected to ground) 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. Indicates the receive code group error. This signal is synchronous to RX_CLK_R and mapped to the on-board LED4. The LED ON condition indicates an error in the received code group.
SPISCLKO, SPISS, and SPISDO	Output	SPI controller signals to interface with the ZL30364 clock generation hardware.
SPISDI	Input	

2.1.1 Mi-V Soft Processor (Ask a Question)

The Mi-V soft processor supports RISC-V processor-based designs. The Mi-V soft processor executes the application from the LSRAM mapped at 0x80000000. It configures the ZL30364 clock generation hardware through the CoreSPI IP and the VSC PHY through CoreTSE_AHB MDIO interface. It also configures the CoreTSE_AHB registers using the AHB interface.

The following figure shows the Mi-V soft processor configuration, where the **Reset Vector Address** is set to 0x8000_0000. In Mi-V processor memory map, the memory range used for the AHB memory interface is 0x7000_0000 to 0x7FFF_FFFF, and the memory range used for the APB interface is 0x6000_0000 to 0x6FFF_FFFF.

Figure 2-3. Mi-V Configuration

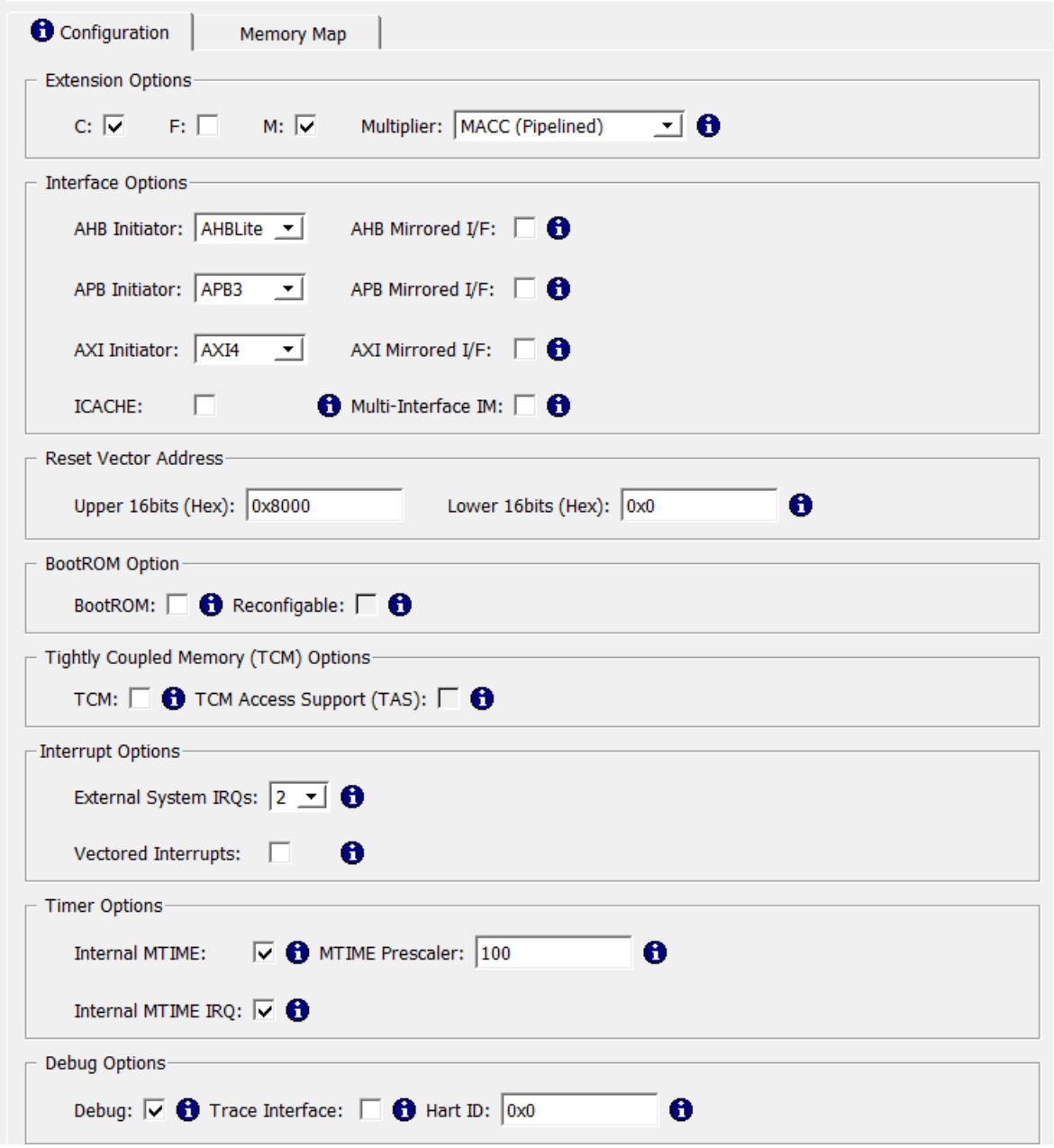
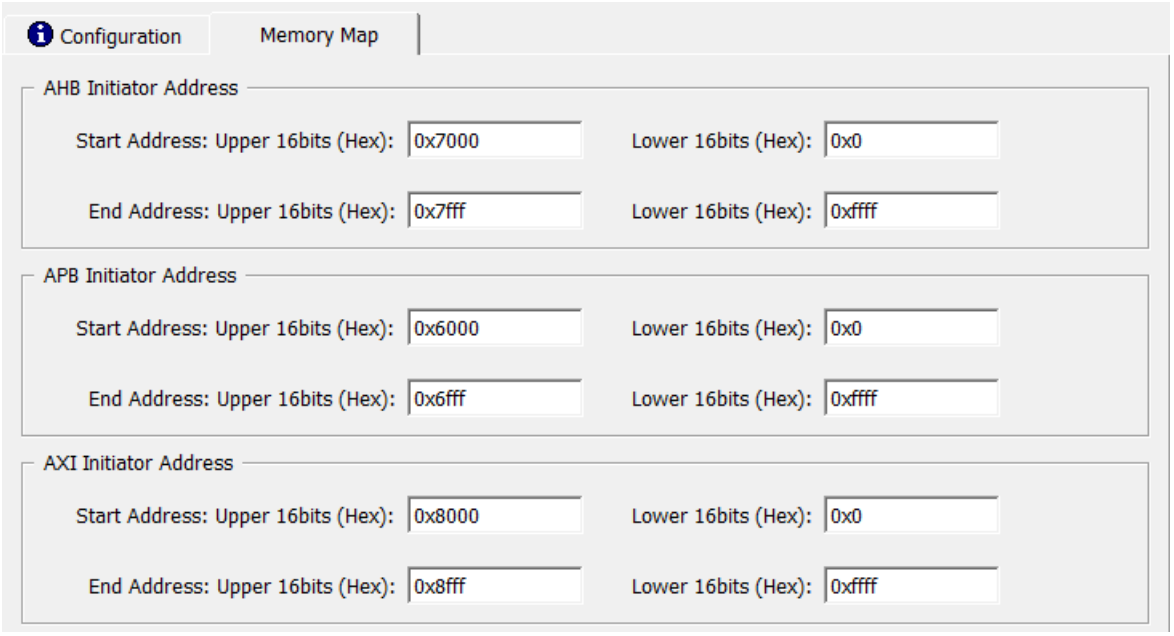


Figure 2-4. Mi-V Memory Map



2.1.2 PF_SRAM_AHBL_AXI (Ask a Question)

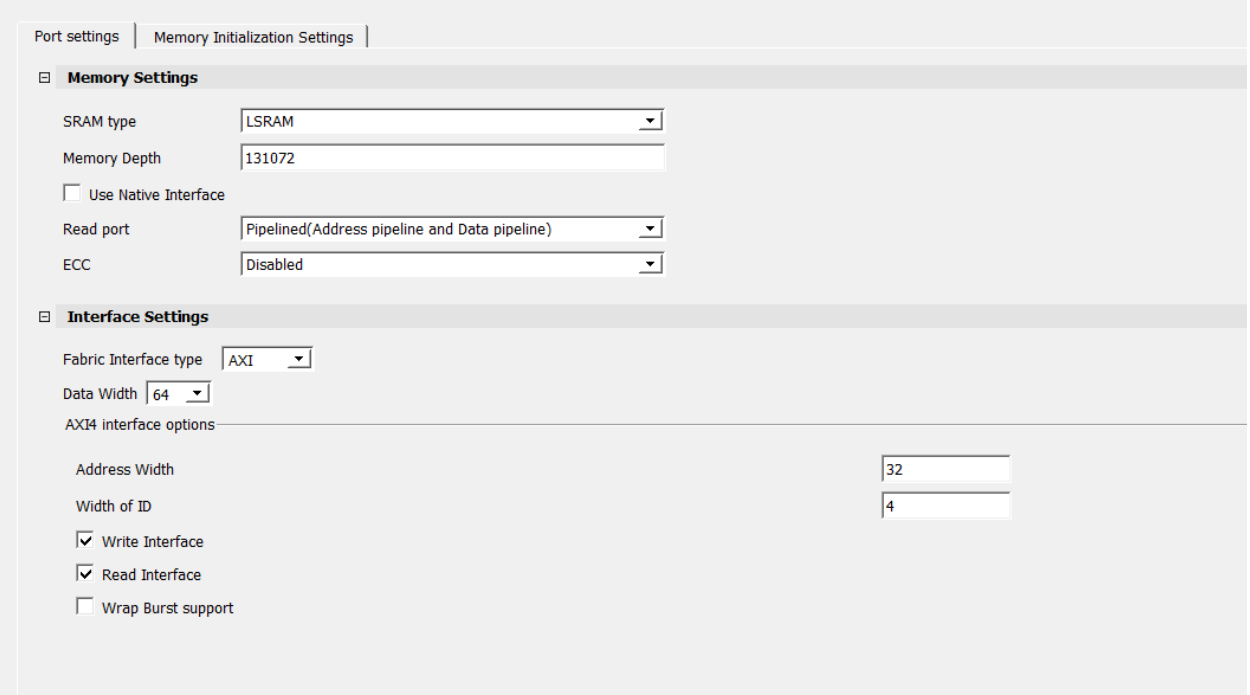
This design uses three instances of PF_SRAM_AHBL_AXI core—pf_sram_0, PF_LSRAM_1_0, and PF_LSRAM_1_1.

The pf_sram_0 IP is connected to Mi-V as an AXI4 slave using Core AXI4Interconnect. The LSRAM blocks are initialized with the user application code from the external SPI Flash.

The processor uses the SRAM memory to execute the application. The following figure shows the LSRAM depth and the interface settings. The **Fabric Interface type** is selected AXI because the fabric interfaces with the Mi-V processor using Core AXI4Interconnect. The memory depth can be selected based on the application size. This design uses 512 KB RAM (131072 words).

The following figure show the PolarFire SRAM configuration.

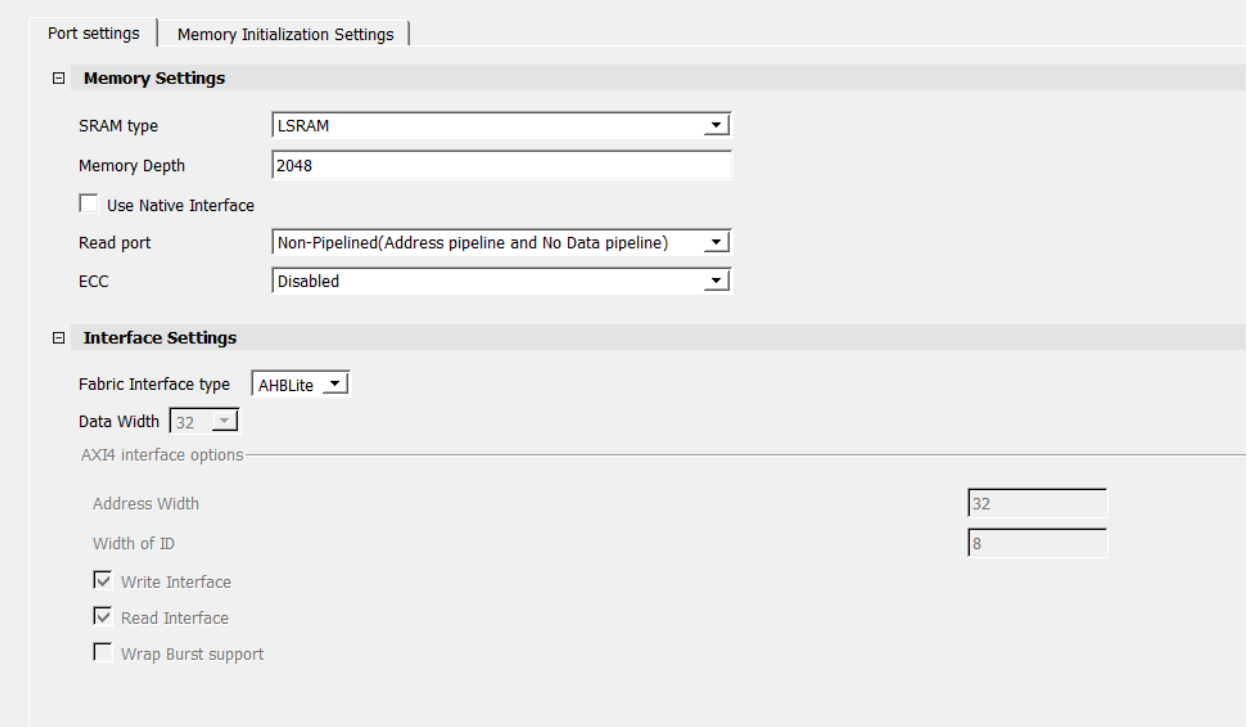
Figure 2-5. PolarFire SRAM_0 Configuration



PF_LSRAM_1_0 and PF_LSRAM_1_1 are connected to the Mi-V AHBLite interface using CoreAHBLite. This memory (LSRAM_1_0 and LSRAM_1_1) is used for Ethernet MAC transmit and receive buffers.

The following figure show the PolarFire LSRAM configuration.

Figure 2-6. PolarFire LSRAM _1_0 Configuration



2.1.3 CoreAXI4Interconnect (Ask a Question)

The AXI interconnect bus must be configured to connect the Mi-V core with memory. The following figure shows the bus configuration and other configuration of CoreAXI4Interconnect.

Figure 2-7. CoreAXI4Interconnect Configurator

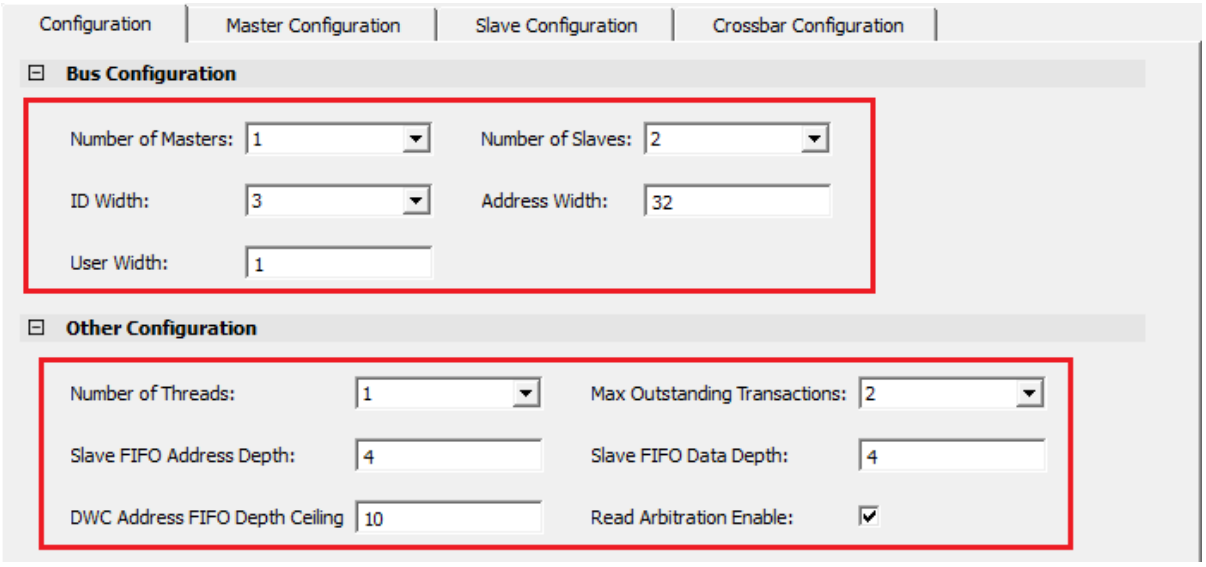


Figure 2-8. CoreAXI4Interconnect Configurator-Master Configuration

Figure 2-9. CoreAXI4Interconnect Configurator-Slave Configuration

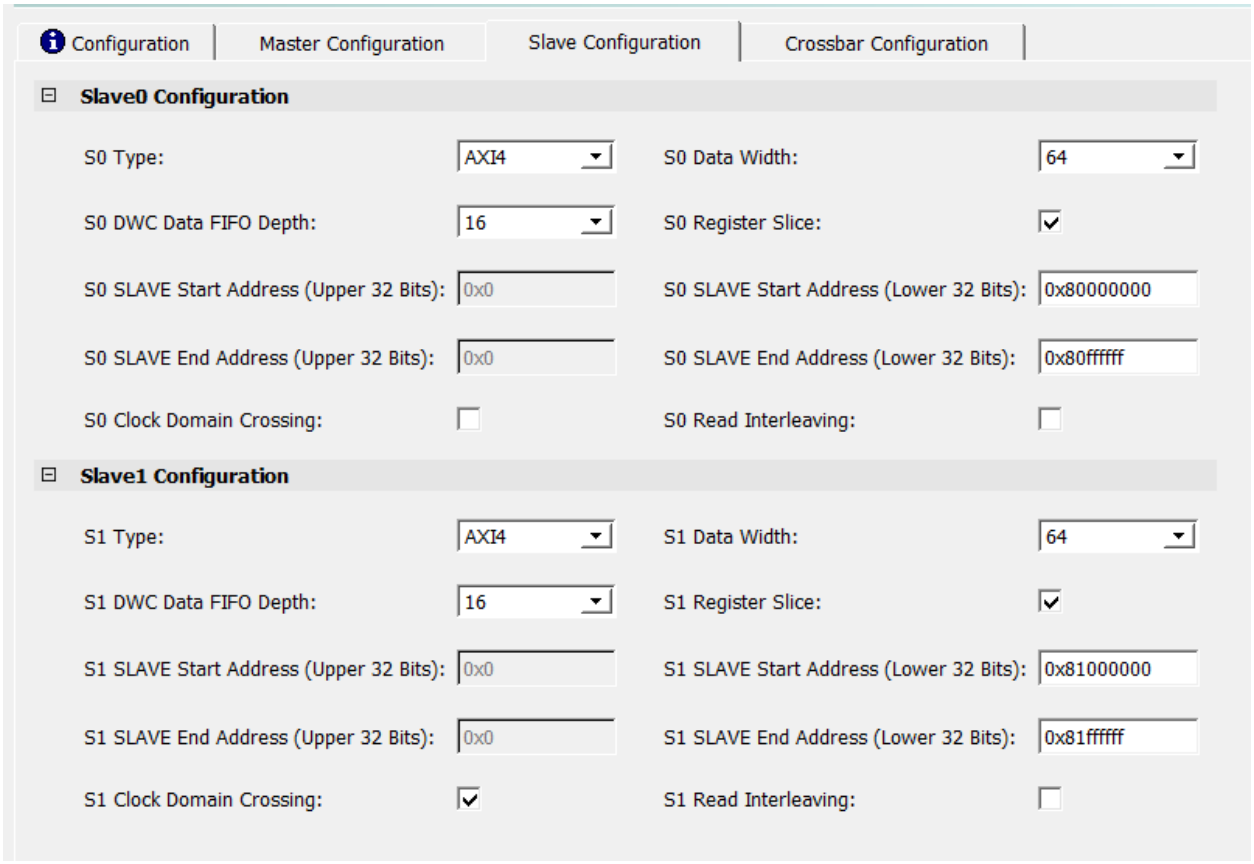
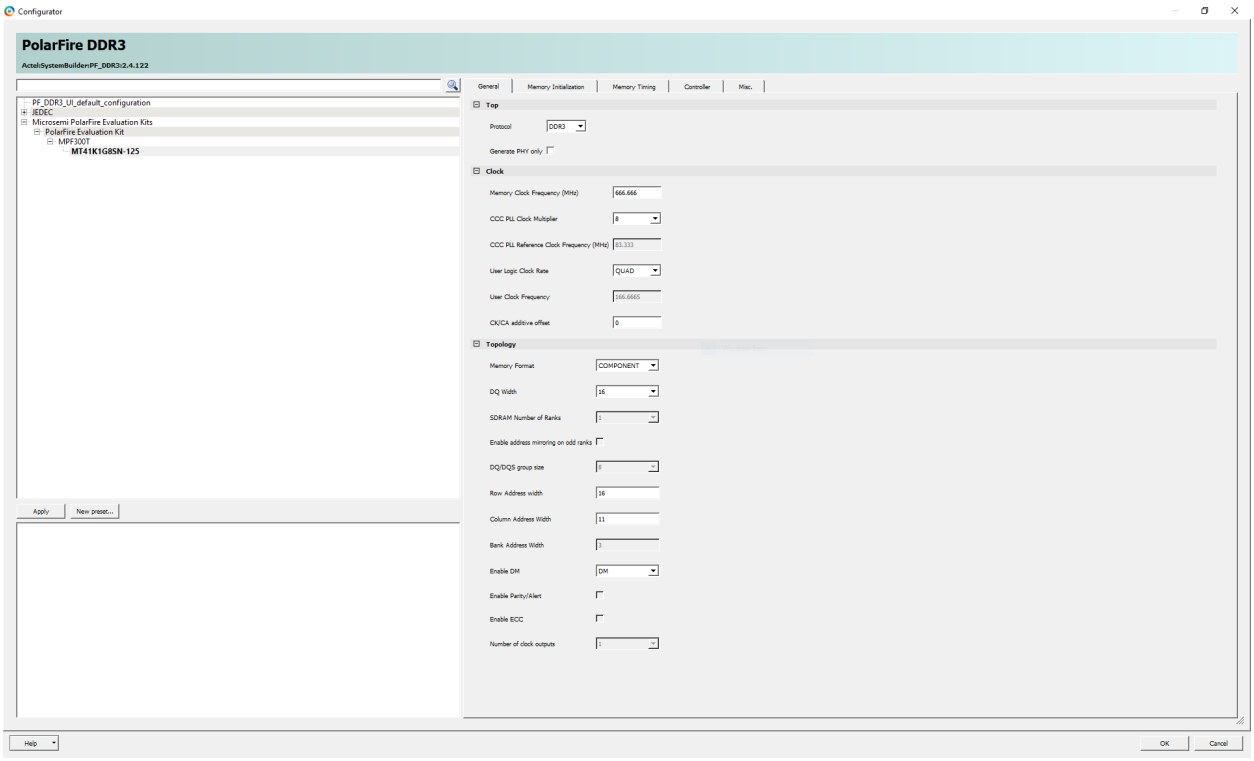


Figure 2-10. CoreAXI4Interconnect Configurator-Crossbar Configuration

2.1.4 DDR3 (Ask a Question)

The DDR3 subsystem is configured to access the 16-bit DDR3 memory through an AXI4 interface. The PolarFire Evaluation's Kit DDR3 memory preset is applied to configure all the memory initialization and timing parameters in the DDR configurator. The following figure shows general configuration settings of the DDR3 memory.

Figure 2-11. DDR3 Configuration

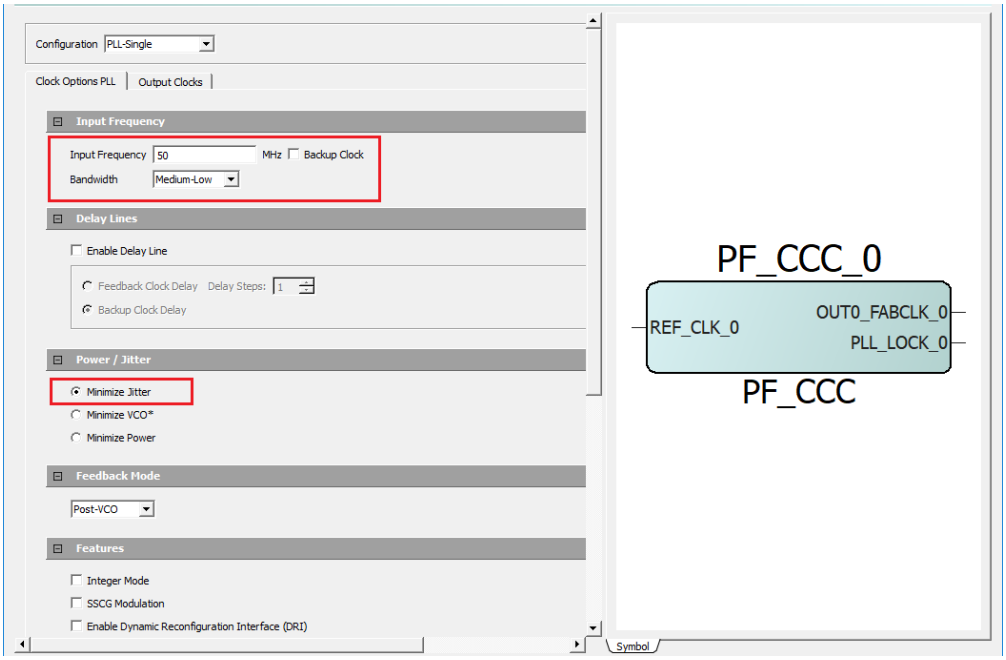


Important: For more information about Rev E or later Kit DDR3 Configurations (MT41K512M8DA-107: P), see *Appendix 3 - DDR3 Configuration* of the AN4997: PolarFire FPGA Building a Mi-V Processor Subsystem.

2.1.5 PF_CCC_0 (Ask a Question)

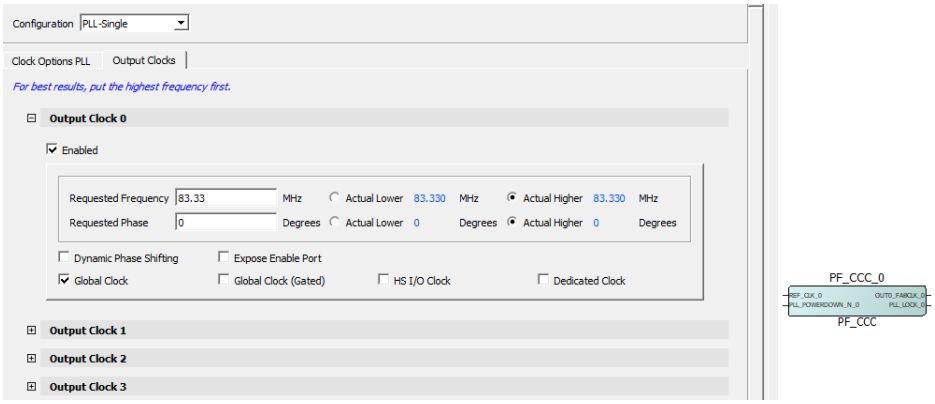
PF_CCC_0 (PolarFire Clock Conditioning Circuitry) generates the fabric reference clock that drives the soft processor and the APB peripherals. The PF_CCC_0 IP is configured to generate one output fabric clock from a 50 MHz input. The following figure shows the PF_CCC_0 input clock configuration.

Figure 2-12. PF_CCC_0 Input Clock Configuration



The following figure shows the PF_CCC_0 output clock configuration. The Mi-V processor supports up to 120 MHz. This design uses an 83.33 MHz system clock for configuring the APB peripherals.

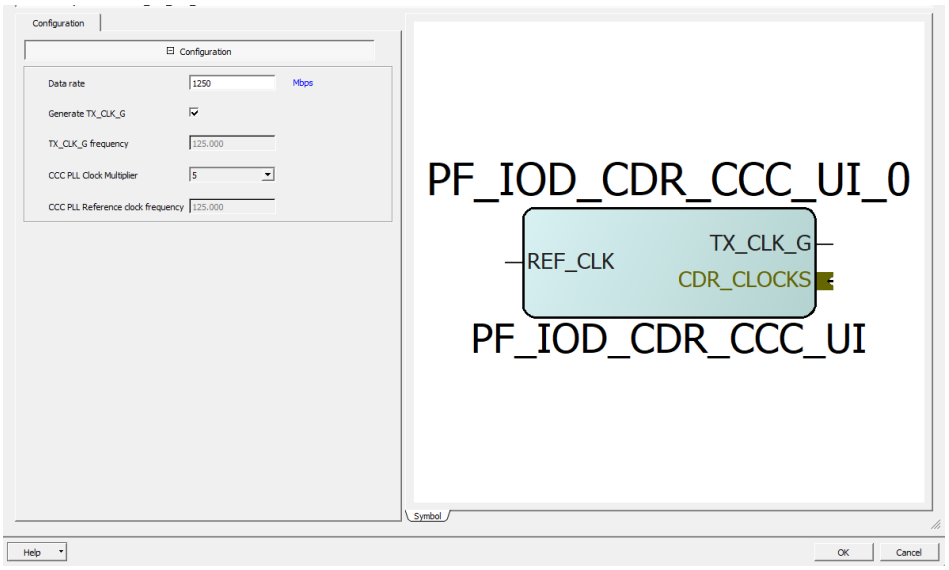
Figure 2-13. PF_CCC_0 Output Clock Configuration



2.1.6 PF_IOD_CDR_CCC (Ask a Question)

The PF_IOD_CDR_CCC block used for IOD CDR. PF_IOD_CDR_CCC is configured to generate output fabric clocks from a 125 MHz input. One clock is required for clock recovery and one clock for the fabric TX interface of the CoreTSE block. The following figure shows the PF_IOD_CDR_CCC input clock configuration.

Figure 2-14. PolarFire IOD CDR Clocking Configuration



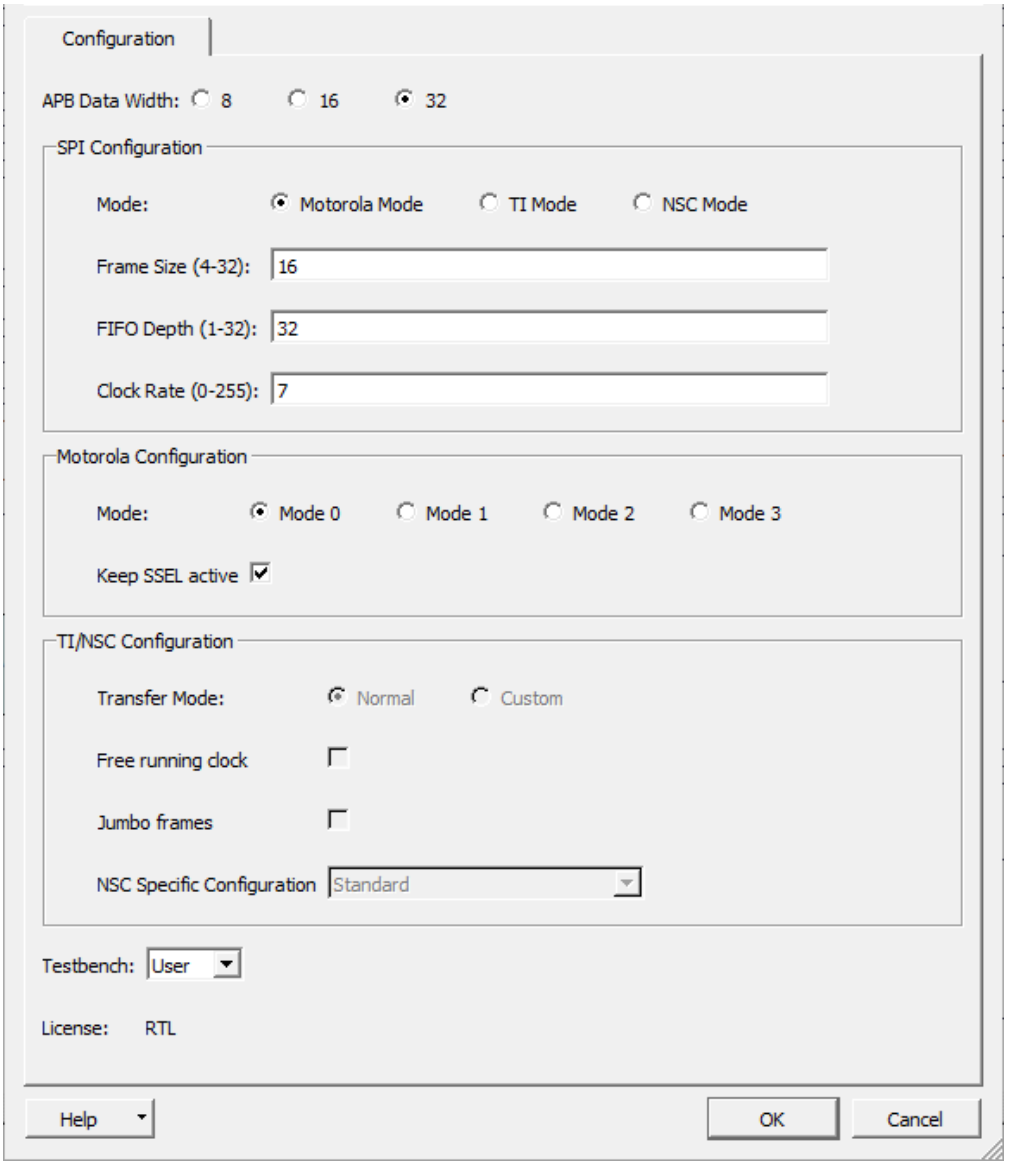
2.1.7 CORESPI_0 (Ask a Question)

The CORESPI0 (CoreSPI) block is a controller IP that implements the serial communication. Mi-V configures the ZL30364 clock generation hardware using the CORESPI_0 block. The recommendations for CoreSPI configuration is provided in the following points, see Figure 2-15.

- **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.
- The clock rate for the SPI master clock is selected as 7. This is used to generate the SPICLK, which is generated as PCLK/(2*(clock rate+1) = 83.33/(2*(7+1)).
- The **Keep SSEL active** check box is enabled to keep the slave peripheral active between back-to-back data transfers.

The following figure shows the CoreSPI configuration.

Figure 2-15. CoreSPI_0 Configuration



2.1.8 Core_SPI_FLASH_0 (Ask a Question)

This core is configured for accessing the external SPI Flash for IAP. The configuration options are same as Figure 2-15.

2.1.9 CoreGPIO_0_0 (Ask a Question)

This core is configured to control the on-board LEDs and switches.

2.1.10 PF_SPI_0 (Ask a Question)

This macro is an interface between the system controller SPI and the CoreSPI controller.

2.1.11 CoreAPB3_0 (Ask a Question)

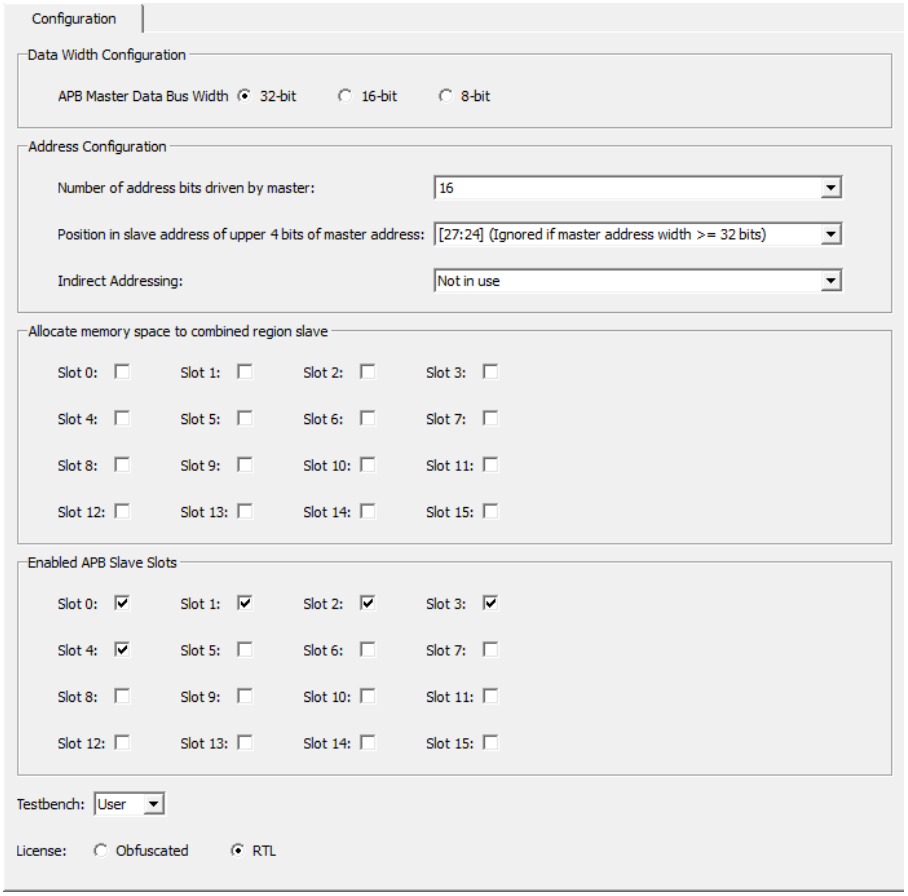
CoreAPB3_0 is configured as shown in the Figure 2-16 to connect the peripherals CoreSPI, Core_SPI_Flash, CoreGPIO, PF_SYSTEM_SERVICES, and CoreUARTapb as slaves.

The following are the features of CoreAPB3 configuration.

- **APB Master Data bus width**: 32 bit
- A number of address bits are driven by the 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, S2, S3, and S4 (for CoreUARTapb, CoreSPI, Core_SPI_Flash, CoreGPIO, and PF_SYSTEM_SERVICES respectively)

The following figure shows the CoreAPB3 configuration.

Figure 2-16. CoreAPB3 Configuration



2.1.12 COREAHBTOAPB3_0 (Ask a Question)

The COREAHBTOAPB3 IP connects to CoreAPB3. This IP retains the default configuration.

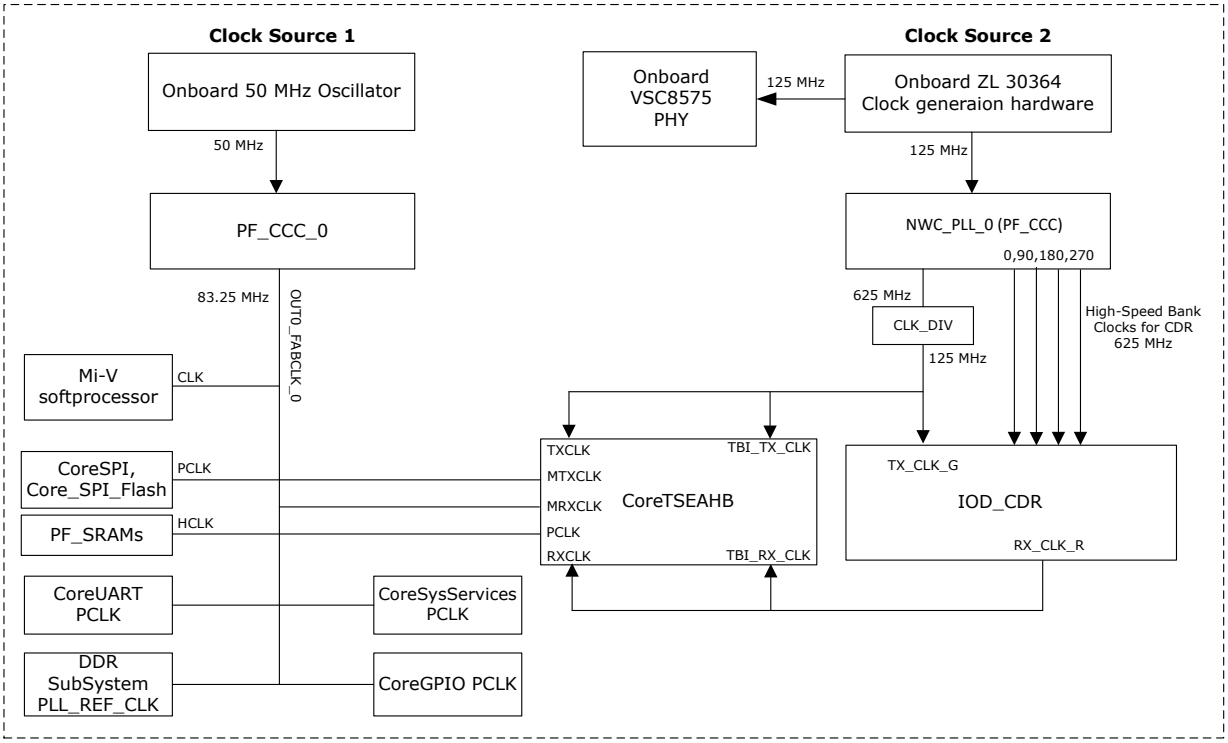
2.2 Clocking Structure (Ask a Question)

In the demo design, there are two clock domains—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 83.33 MHz clock for the Mi-V soft processor and peripherals. The Mi-V soft processor can operate at maximum 120 MHz. In this design, the Mi-V processor runs at 83.33 MHz.
- On-board ZL 30364 clock generation hardware: This hardware generates the reference clocks for the VSC PHY and the IOD CDR fabric module.

The following figure shows the clocking structure of the demo design.

Figure 2-17. Clocking Structure



3. Demo Requirements (Ask a Question)

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

Table 3-1. Demo Requirements

Requirement	Version
Hardware	
PolarFire Evaluation Kit (MPF300-EVAL-KIT)	Rev D or later
Ethernet cable	RJ45
Software	
FlashPro Express	Refer to the readme.txt file provided in the design files for the software versions used with this reference design.
Libero SoC	
SoftConsole	
Serial terminal emulation program	HyperTerminal, Tera Term, or Latest version of PuTTY 0.76
Browser	Mozilla Firefox, Internet Explorer

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

4. Demo Prerequisites (Ask a Question)

Before you begin:

1. Download demo design files from this link:
   www.microchip.com/en-us/application-notes/AN4569
2. Download and install Libero SoC (as indicated in the website for this design) on the host PC from the following location:
   Libero SoC Installation link

   The latest versions of ModelSim® and Synplify Pro® are included in the Libero SoC installation package.

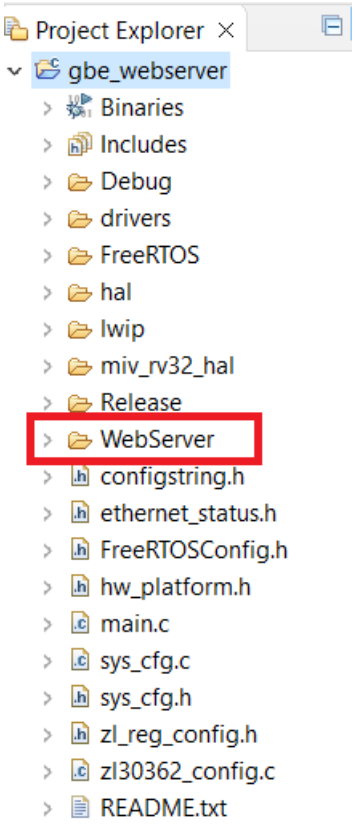
5. SoftConsole Firmware Project (Ask a Question)

The following stacks available in the SoftConsole **Project Explorer** are used in this demo design.

- lwIP TCP/IP stack v1.4.1
- FreeRTOS

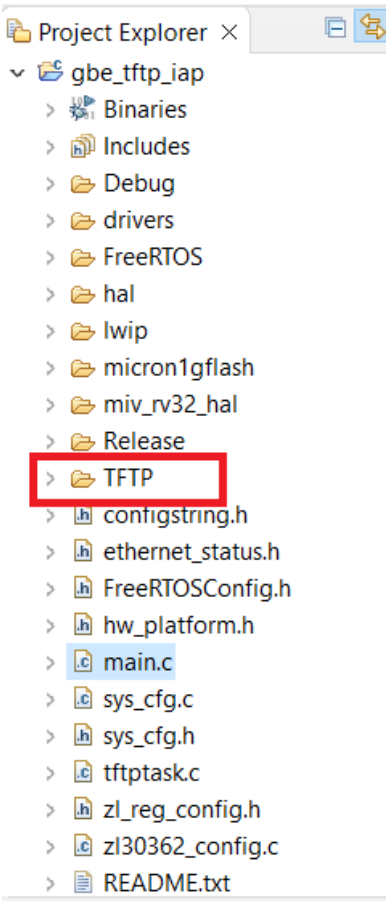
The following figure shows the directory structure of the webserver SoftConsole project (located at `<$design_file_directory>\mpf_an4569_v2023p2_eval_df\webserver\SoftConsole_Project`). It contains the webserver application (which uses LWIP and FreeRTOS) and all the firmware and hardware abstraction layers that correspond to the hardware design.

Figure 5-1. Directory Structure of Webserver SoftConsole Project



The following figure shows the directory structure of the TFTP_IAP SoftConsole project (located at `<$design_file_directory>\mpf_an4569_v2023p2_eval_df\tftp_iap\SoftConsole_Project`). It contains the IAP application, TFTP server (which uses LWIP and FreeRTOS), and all the firmware and hardware abstraction layers that correspond to the hardware design.

Figure 5-2. Directory Structure of TFTP IAP Softconsole Project



6. Libero Design Flow (Ask a Question)

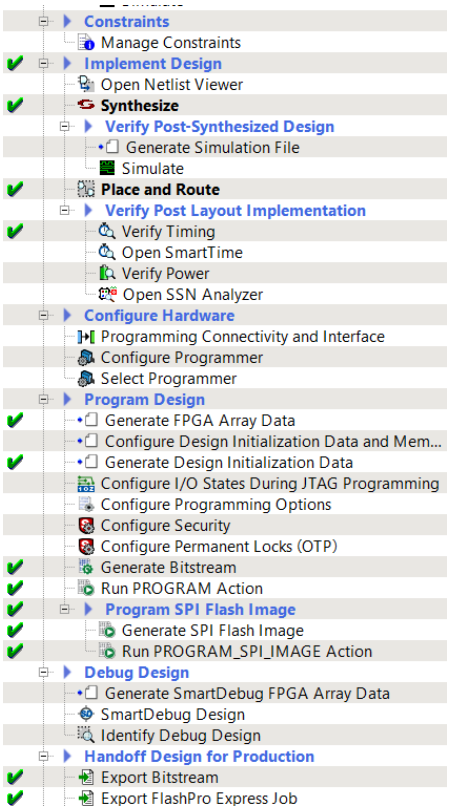
This chapter describes the Libero design flow for running this demo design, which includes:

- 6.1. Synthesize
- 6.2. Place and Route
- 6.3. Verify Timing
- 6.4. Generate FPGA Array Data
- 6.5. Configure Design Initialization Data and Memories
- 6.6. Generate Bit stream
- 6.7. Export FlashPro Express Job
- 6.8. Run PROGRAM Action
- 6.9. Program SPI Flash Image

For more information about creating the Libero project, see 11. Appendix 4: Running the TCL Script.

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

Figure 6-1. Libero Design Flow Options



6.1 Synthesize (Ask a Question)

To synthesize the design, perform the following steps:

1. On the **Design Flow** tab, double click **Synthesize**.
   When the synthesis is successful, a green tick mark appears next to Synthesize, as shown in the Figure 6-1.

2. Right click **Synthesize** and select **View Report** to view the synthesis report and log files in the **Reports** tab.

6.2 Place and Route (Ask a Question)

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

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

1. On the **Design Flow** tab, double click **Place and Route**.

   When place and route is successful, a green tick appears next to **Place and Route**, as shown in Figure 6-1.
2. Right click **Place and Route** and select **View Report** to view the place and route report and log files in the **Reports** tab.

6.2.1 Resource Utilization (Ask a Question)

The following table lists the resource utilization of the design after place and route. These values may vary slightly for different Libero runs, settings, and seed values.

Table 6-1. Resource Utilization

Type	Used	Total	Percentage
4LUT	58856	299544	19.65
DFF	44408	299544	14.83
I/O register	0	1536	0.00
User I/O	87	512	16.99
Single-ended I/O	75	512	14.65
Differential I/O pairs	6	256	2.34

6.3 Verify Timing (Ask a Question)

To verify timing, on the **Design Flow** tab, double click **Verify Timing**.

When the design successfully meets the timing requirements, a green tick appears next to **Verify Timing**, as shown in Figure 6-1.

6.4 Generate FPGA Array Data (Ask a Question)

To generate FPGA array data, on the **Design Flow** tab, double click **Generate FPGA Array Data**.When the FPGA array data is successfully generated, a green tick appears next to **Generate FPGA Array Data**, as shown in Figure 6-1.

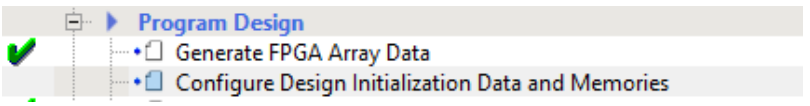
6.5 Configure Design Initialization Data and Memories (Ask a Question)

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

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

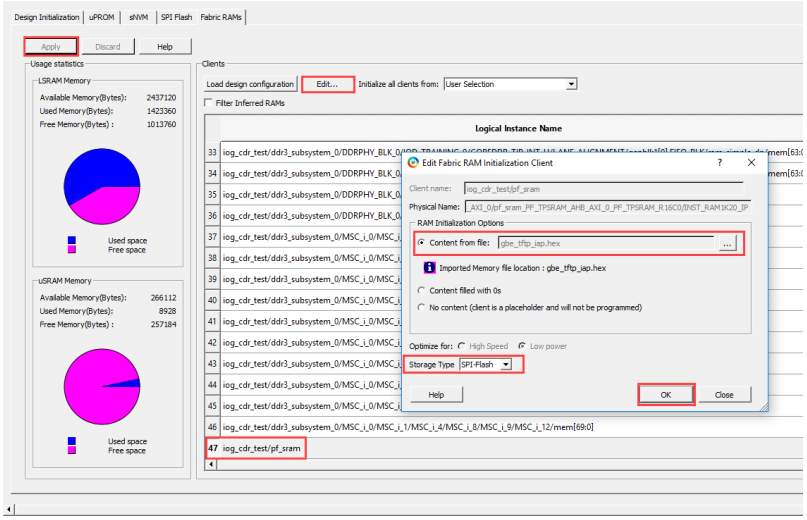
1. On the **Design Flow** tab, double click **Configure Design Initialization Data and Memories**, as shown in the following figure.

Figure 6-2. Configure Design Initialization Data and Memories Option



2. In the **Configure Design Initialization Data and Memories** window, select the **Fabric RAMs** tab, and then select the **pf_sram** file to import the memory information, as shown in the following figure.

Figure 6-3. Fabric RAMs Tab



3. Import the hex file (`gbe_webserver.hex` for webserver design and `gbe_tftp_iap.hex` for TFTP_IAP design) provided with the design files from

```
mpf_an4569_v2023p2_eval_df\webserver\TCL_Scripts\src\src_softconsole\gbe_webserver.hex
```

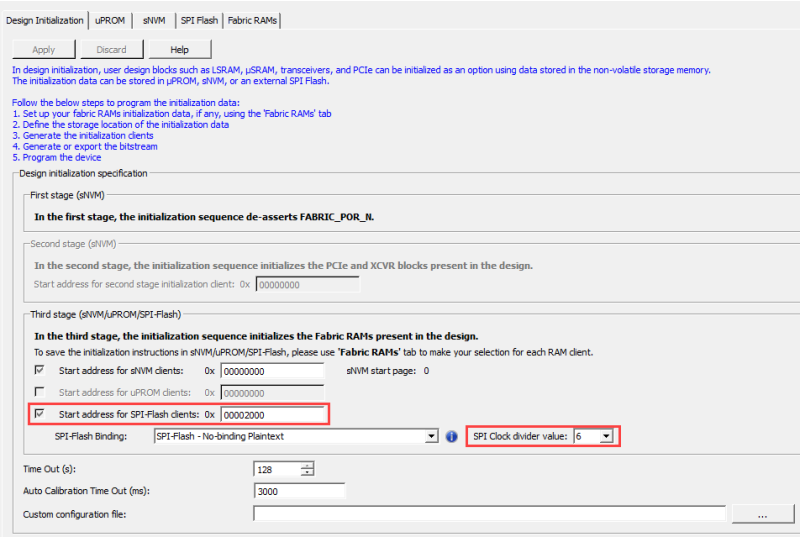
or

```
mpf_an4569_v2023p2_eval_df\tftp_iap\TCL_Scripts\src\src_softconsole\gbe_tftp_iap.hex
```

The `gbe_webserver.hex` or `gbe_tftp_iap.hex` file is a application file generated using SoftConsole that configures the ZL clock generation hardware, the CoreTSE_AHB registers, and the VSC PHY. The application code is initially stored in an external SPI Flash. On device power-up, the system controller copies the code to LSRAM from external SPI Flash, and the Mi-V processor executes the code from LSRAM. To ensure that the fabric LSRAM contents are stored in the external SPI Flash, select **Storage Type as SPI Flash**, see in Figure 6-3.

4. Click **Apply**.
5. Select **Start address for SPI Flash clients** and **SPI Clock divider value**, in the **Design Initialization** tab as shown in the following figure.

Figure 6-4. Start Address for SPI Flash Clients



Note: The default start address for SPI-Flash clients 0x400 is used for the webserver design. The start address for TFTP design is modified to 0x2000. This is required to support Flash erase of 4 KB while writing the SPI directory into initial SPI Flash 1 KB memory using design firmware.

6. On the **Design Flow** tab, double click **Generate Design Initialization Data**.
   When the LSRAM initialization client is successfully generated in sNVM, a green tick appears next to **Generate Design Initialization Data**, as shown in Figure 6-1.
   When the device is programmed, the LSRAM block is initialized from the sNVM.

6.6 Generate Bit stream (Ask a Question)

To generate the bit stream, perform the following steps:

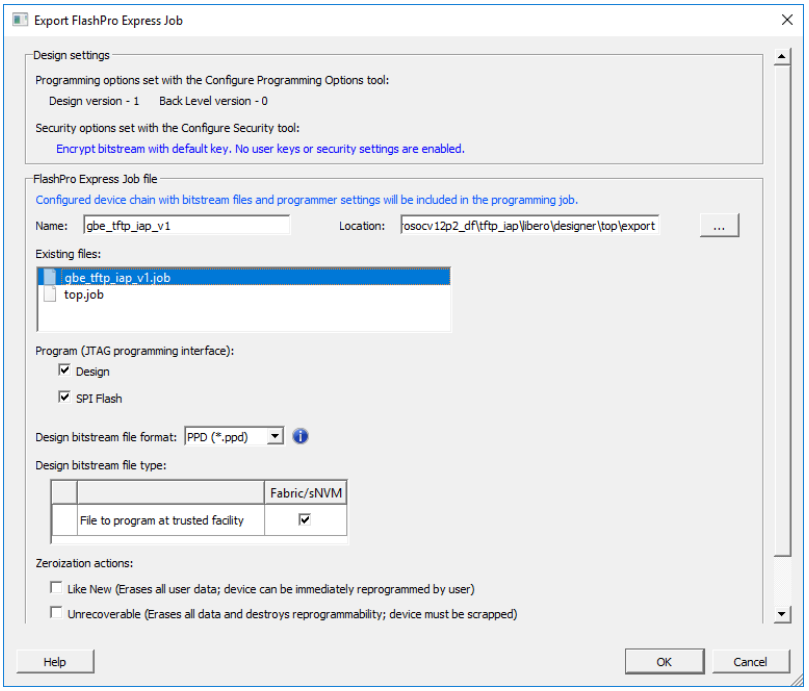
1. On the **Design Flow** tab, double click **Generate Bit stream**.
   When the bit stream is successfully generated, a green tick appears next to **Generate Bit stream**, as shown in Figure 6-1.
2. Right click **Generate Bit stream** and select **View Report** to view the corresponding log file in the **Reports** tab.

6.7 Export FlashPro Express Job (Ask a Question)

To generate `.job` file, perform the following:

On the **Design Flow** tab, double click **Export FlashPro Express Job** and select Design and SPI Flash as shown in figure. The exported job file contains the data contents to be programmed into PolarFire FPGA and external SPI Flash. This job file is used in the FlashPro Express software to program both device and external SPI Flash as shown in 10. Appendix 3: Programming the Device and External SPI FlashPro Express.

Figure 6-5. Export FlashPro Express Job



6.8 Run PROGRAM Action (Ask a Question)

After generating the bit stream, the PolarFire device must be programmed. Follow these steps to program the PolarFire device using the Libero design flow:

Note: If you want to program the PolarFire FPGA using the `.job` file instead, see 10. Appendix 3: Programming the Device and External SPI FlashPro Express.

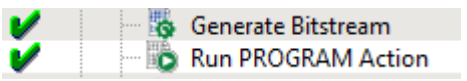
1. Ensure that the jumper settings on the board are as listed in the following table.

Table 6-2. 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 2.5V

2. Connect the power supply cable to the J9 connector on the board.
3. Connect the USB cable from the host PC to J5 Future Technology Devices International Limited (FTDI) port on the board.
4. Connect any one of the open network 1G Ethernet capable ports to the J15 connector (RJ45-PORT 0) on the board.
5. Power-up the board using the SW3 slide switch.
6. On the Libero Design Flow tab, double click **Run PROGRAM Action**.
   When the device is successfully programmed, the LEDs 6 and 7 on the board glow, and a green tick appears, as shown in the following figure.

Figure 6-6. Run Program Action



7. Right click **Run Program Action** and select **View Report** to view the corresponding log file in the **Reports** tab.

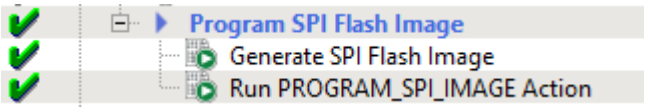
6.9 Program SPI Flash Image (Ask a Question)

To program SPI Flash Image, perform the following steps:

1. Double click **Generate SPI Flash Image** and right click **Run PROGRAM_SPI_IMAGE Action** to get the SPI Flash programmed with the application as shown in the following figure.

Important: If you want to program the external SPI Flash using the `.job` file instead, see Appendix: Programming the Device Using FlashPro Express, page 41.

Figure 6-7. SPI Flash Programming



2. Power-cycle the board after you program the PolarFire device and the external SPI Flash.

The demo is ready to be run. For information about how to run the demo, see 7. Running the Demo.

7. Running the Demo (Ask a Question)

To run the demo design, perform the following steps:

1. Download demo design files from the link: FPGA Documentation.
2. Power-up the board using the SW3 slide switch.
3. Start a serial terminal emulation program such as HyperTerminal, PuTTY, or Tera Term.

Important: For this demo, Tera Term is used.

For more information about configuring serial terminal emulation programs, see Configuring Serial Terminal Emulation Programs Tutorial.

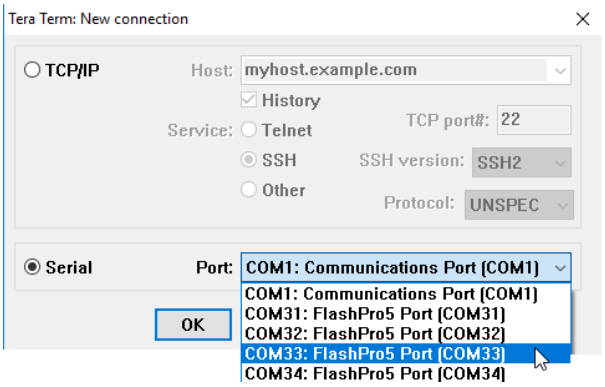
7.1 Tera Term Setup (Ask a Question)

The user application provides a user interface on the Tera Term terminal through the UART interface.

To setup the Tera Term program, perform the following steps:

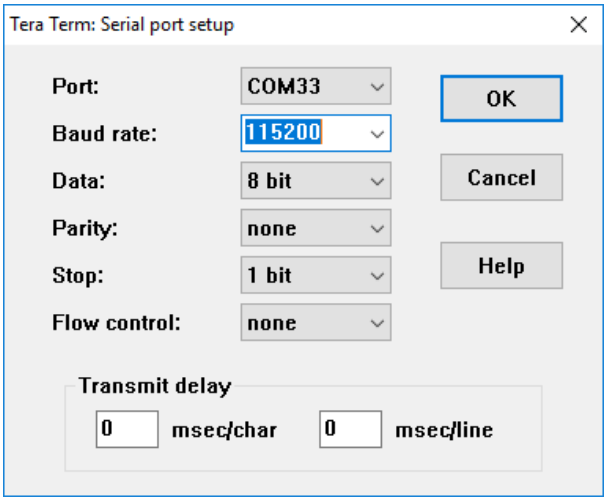
1. Ensure that the USB cable connects the host PC to the **J5** (USB) port on the PolarFire Evaluation board.
2. Start Tera Term.
3. Select **Serial** as the Connection type.
4. Set the serial **Port** to the second highest COM port number from the drop-down list as shown in the following figure. For example, **COM33: FlashPro5 Port [COM33]** in this instance.

Figure 7-1. Select Serial as the Connection Type



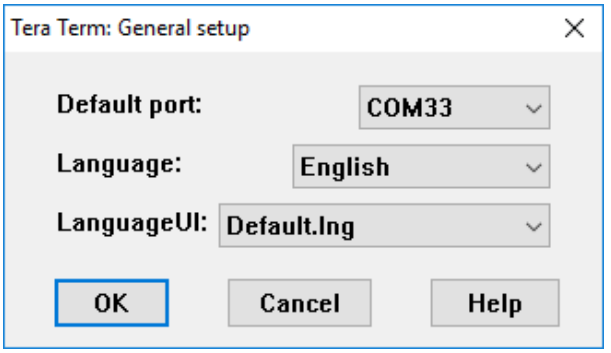
5. In the **Tera Term** window, go to **Setup > Serial port**..., set **Baud rate** to 115200.
   Rest of the serial port settings must be at default state as shown in the following figure and click **OK**.

Figure 7-2. Tera Term Configuration



6. In the **Tera Term** window, go to **Setup > General**..., set the **Language** to **English** and click **OK**, as shown in the following figure. This setup is required for running the Tera Term macro script.

Figure 7-3. Tera Term General Setup



This completes the Tera Term program setup.

7.2 Running Webserver Demo (Ask a Question)

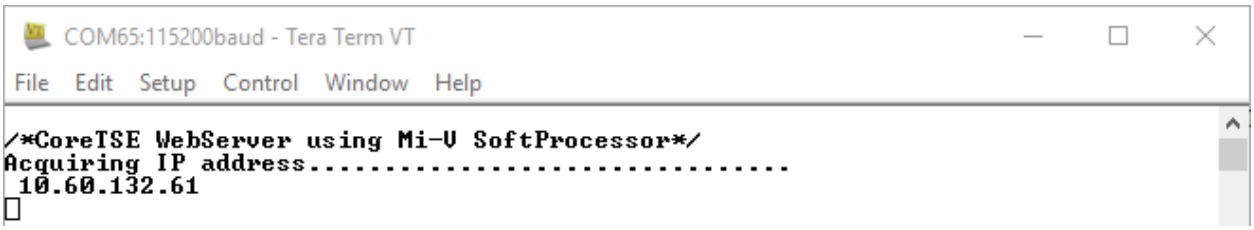
This section describes how to run the Webserver. The following procedure assumes that the serial terminal is setup. For more information about setting up the serial terminal, see 7.1. Tera Term Setup.

Before you begin, perform these steps:

1. Connect the power supply cable to the J9 connector on the board.
2. Connect the USB cable from the host PC to J5 (FTDI port) on the board.
3. Open pin 1 and 2 of the J23 jumper.
4. Connect any one of the open network 1G Ethernet capable ports to the J15 connector (RJ45-PORT 0) on the board.
5. Power-up the board using the SW3 slide switch.
6. Ensure that the device is programmed with the `gbe_webserver.job` file and external SPI Flash is programmed with the application. See 6.9. Program SPI Flash Image to program the external SPI Flash.

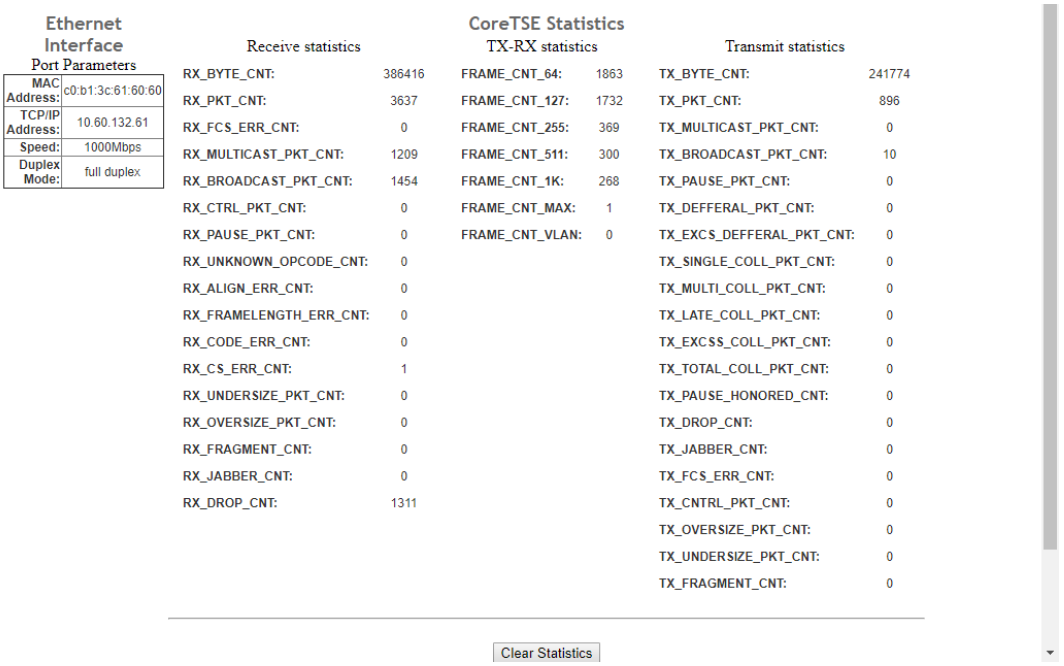
After the device is programmed, power cycle the board. The application prints a welcome message with an IP address on the Tera Term program through the UART interface, as shown in following figure.

Figure 7-4. Tera Term with IP Address



Open a web browser, and enter the IP address displayed on the address bar of the browser. The PolarFire webserver demo page appears, as shown in the following figure. To use the design in static IP mode, see 9.1. Running the Design in Static IP Mode.

Figure 7-5. Webserver Demo Page



7.3 Running TFTP Demo (Ask a Question)

This section describes how to run the IAP using TFTP. The following procedure assumes that the serial terminal is setup. For more information about setting up the serial terminal, see 7.1. Tera Term Setup.

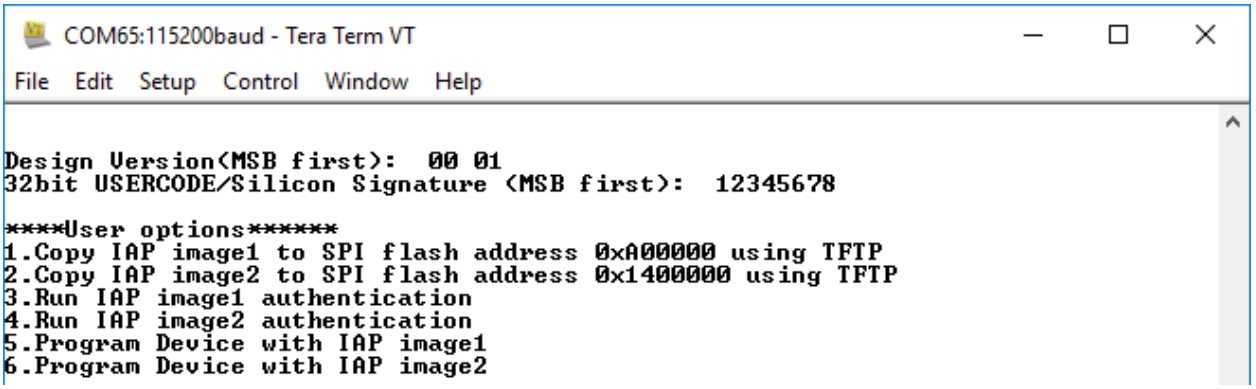
Before you begin, perform the following steps:

1. Connect the power supply cable to the J9 connector on the board.
2. Connect the USB cable from the host PC to J5 Future Technology Devices International Limited (FTDI) port on the board.
3. Open pin 1 and 2 of the J23 jumper.

4. Power-up the board using the SW3 slide switch.
5. Ensure that the device is programmed with the `gbe_tftp_iap_v1.job` file and the external SPI Flash is programmed with the application. See 6.9. Program SPI Flash Image to program the external SPI Flash.
6. Enable TFTP client in Host PC. To enable the TFTP client in Host PC, see 8. Appendix 1: Enable TFTP Client.

After power-up, Tera Term displays the options as shown in the following figure. Observe the design version **01** in the device.

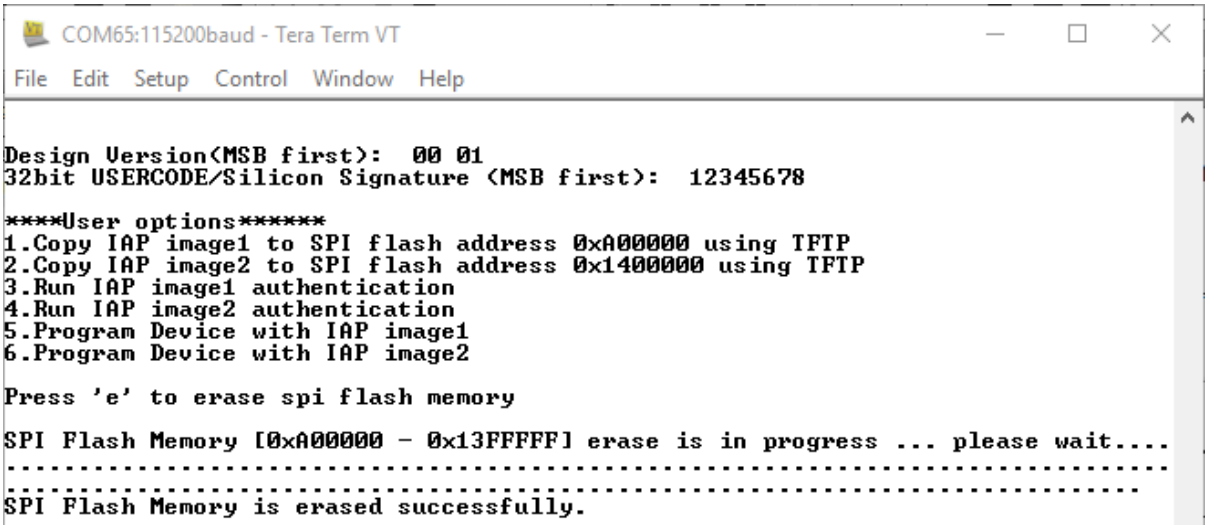
Figure 7-6. Tera Term Window



After power-up, perform the following steps:

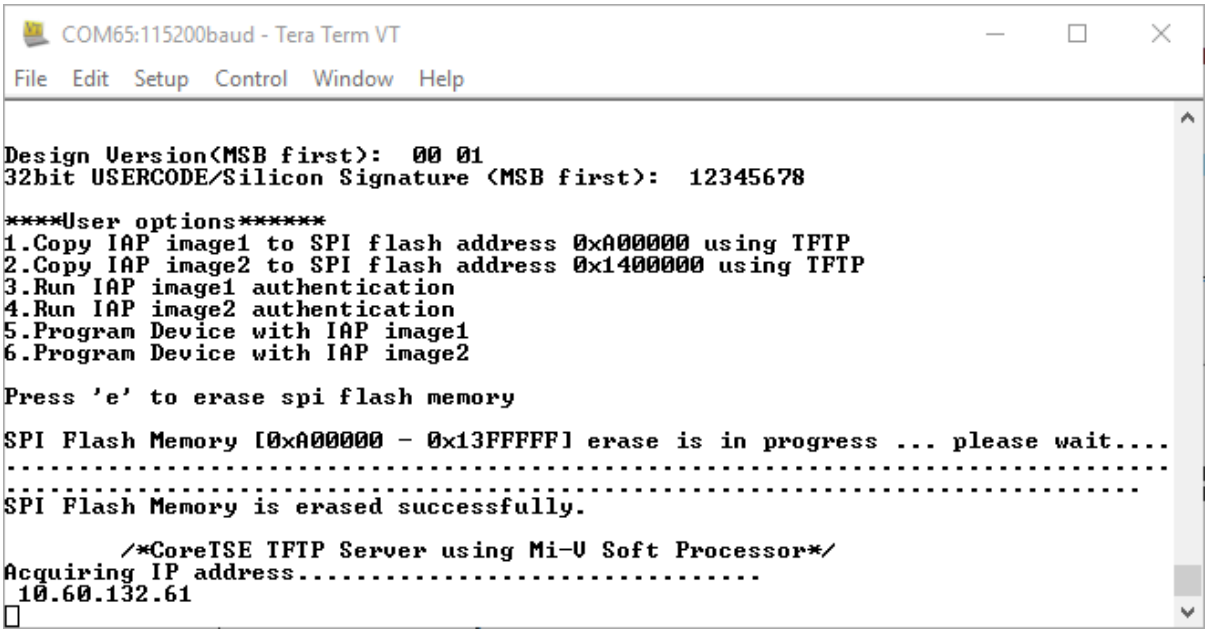
1. Press **1** to load IAP Image1 to SPI Flash address 0xA00000 using TFTP.
2. Press **e** to erase the SPI Flash memory location (0xA00000 - 0x13FFFFF).

Figure 7-7. Erasing SPI Flash Memory Location [0xA00000 - 0x13FFFFF]



3. After completion of the SPI Flash erase operation, the Ethernet link is up, and the IP address is displayed on the Tera Term terminal. In this example, the IP address is 10.60.132.61. The TFTP command uses this IP address to transfer the file to the external SPI Flash. The LED G1 on the PolarFire Evaluation Kit board starts blinking. To use the design in static IP mode, see 9.1. Running the Design in Static IP Mode.

Figure 7-8. Acquiring IP Address

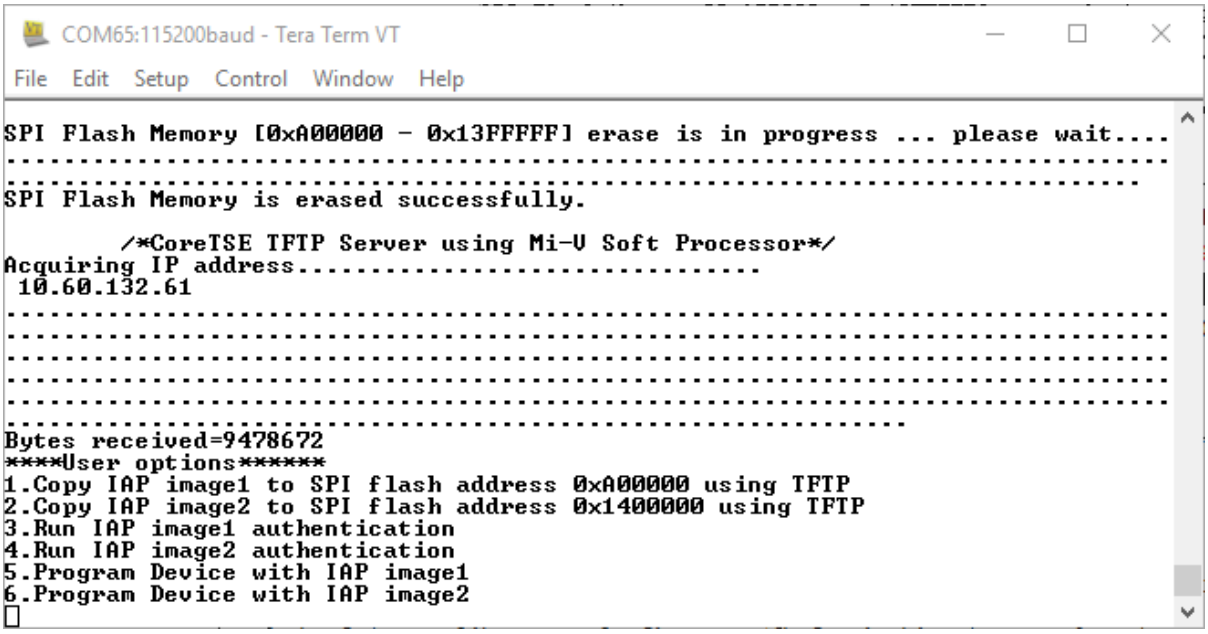


4. On the Host PC command prompt, browse to the folder `<$design file directory>\mpf_an4569_v2023p2_eval_df\tftp_iap\iap_images`.
5. Type the `tftp -i 10.60.132.61 PUT iog_cdr_tftp_iap_v2.spi` command to transfer the `iog_cdr_tftp_iap_v2.spi` programming file to the SPI Flash as shown in the following figure.

Figure 7-9. Transfer Programming Image1

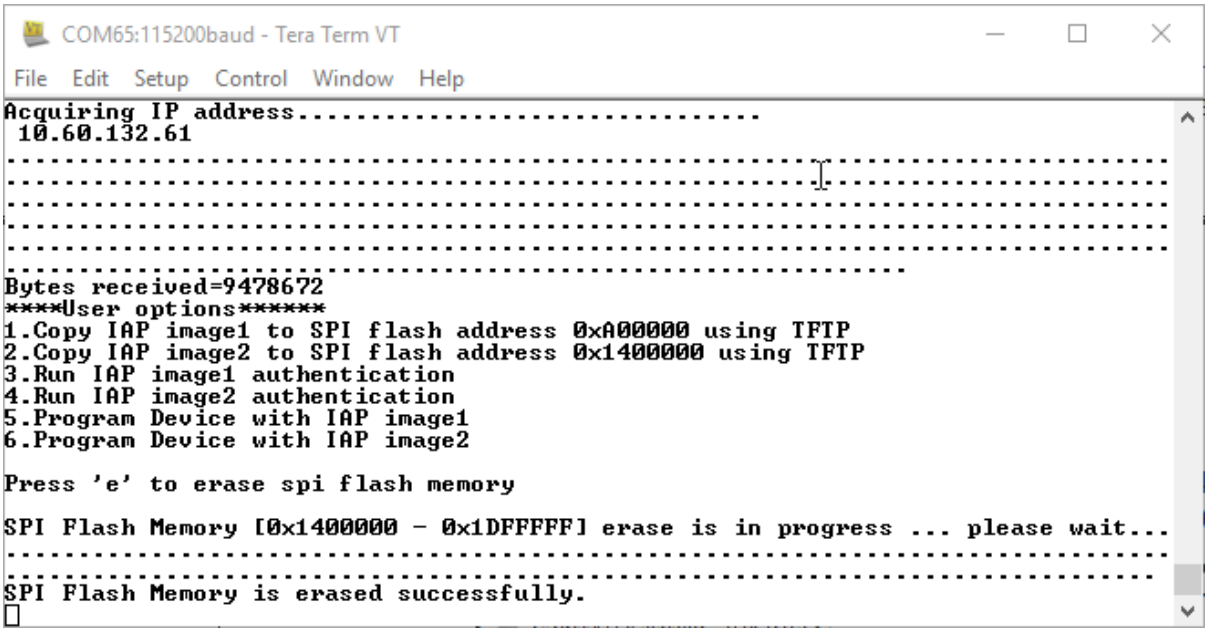
6. Wait until total bytes received message to ensure that the programming image1 is transferred to the SPI Flash. On completion of the Image1 transfer, the user options are displayed.

Figure 7-10. Bytes Received for Image1



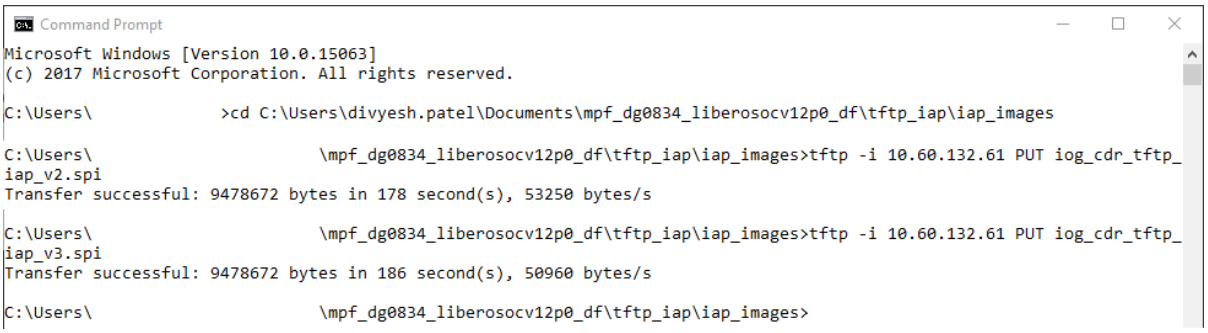
7. Press **2** to load IAP Image2 to SPI Flash address 0x1400000 using TFTP.
8. Press **e** to erase the SPI Flash memory location (0x1400000 - 0x1DFFFFF).

Figure 7-11. Erasing the SPI Flash Memory Location [0x1400000 - 0x1DFFFFF]



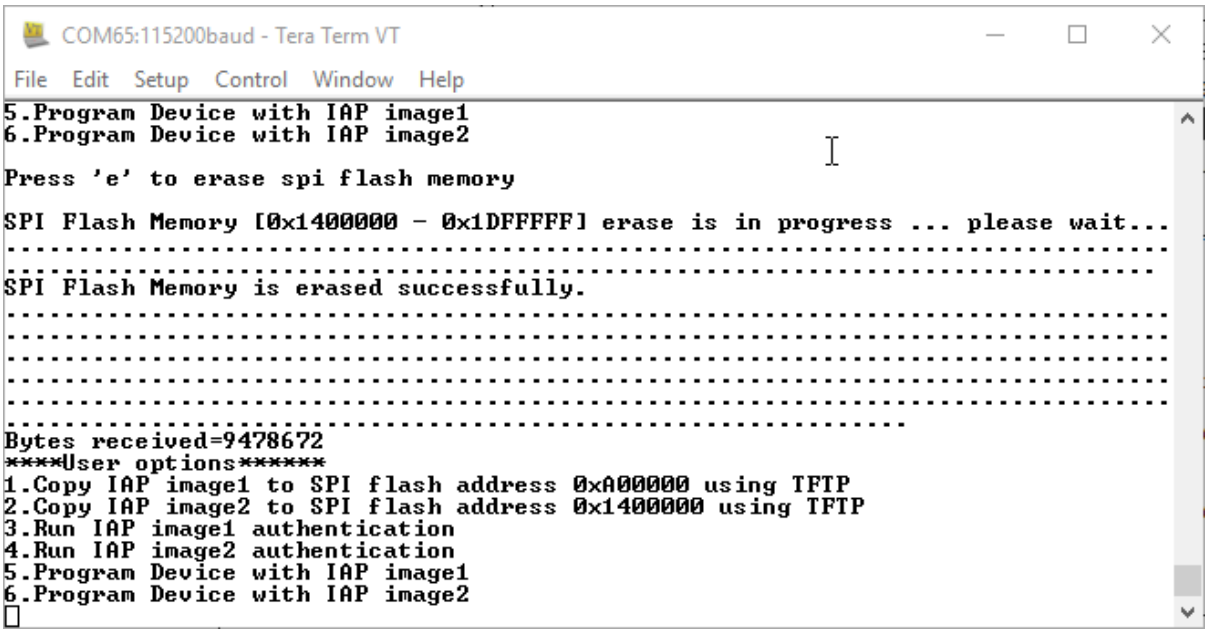
9. On the Host PC command prompt, ensure to browse the folder `<$design file directory>\mpf_an4569_v2023p2_eval_df\tftp_iap\iap_images`.
10. Type the `tftp -i 10.60.132.61 PUT iog_cdr_tftp_iap_v3.spi` command to transfer the programming image2 as shown in the following figure.

Figure 7-12. Transfer IAP Image2



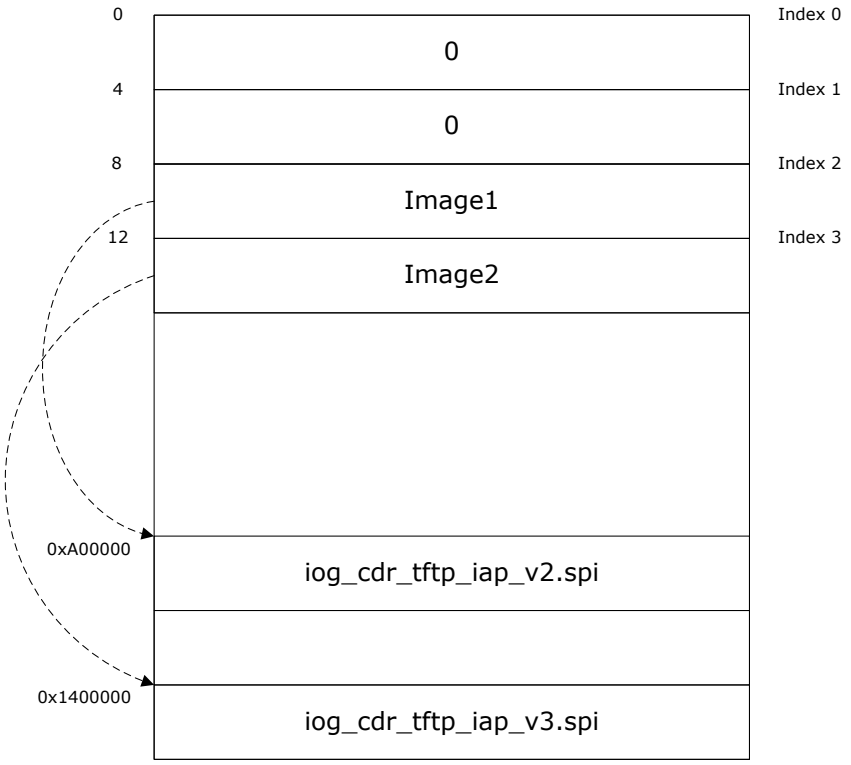
11. Wait until total bytes received message to ensure that the programming image2 is transferred to the SPI Flash. On completion of the Image2 transfer, the user options are displayed.

Figure 7-13. Bytes Received for Image2



The images are transferred successfully to external SPI Flash using TFTP. The firmware application takes care of the external SPI Flash programming with SPI directory as shown in the following figure.

Figure 7-14. SPI Directory

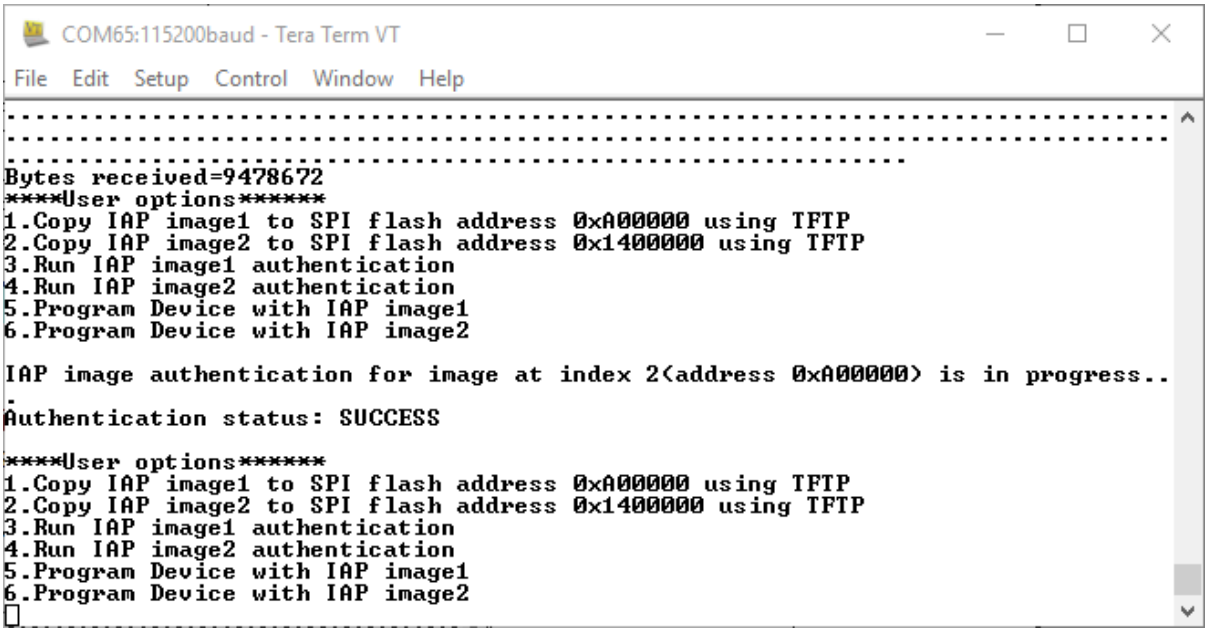


7.3.1 Running IAP Authentication (Ask a Question)

To run the IAP authentication, perform the following steps:

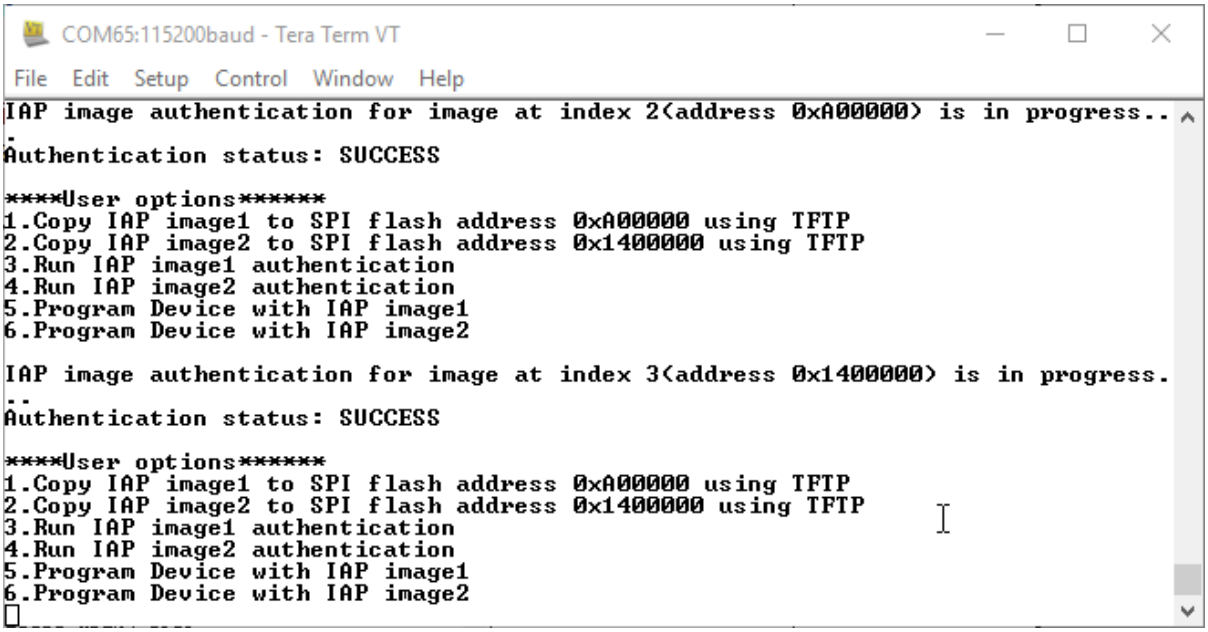
1. Press 3 to initiate the IAP image1 authentication. The IAP authentication with image at index 2 is executed successfully. Tera Term displays the status code as shown in the following figure.

Figure 7-15. Successful IAP Image1 Authentication



2. Press 4 to initiate the IAP image2 authentication. The IAP authentication with image at index 3 is executed successfully. Tera Term displays the status code, as shown in the following figure.

Figure 7-16. Successful IAP Image2 Authentication



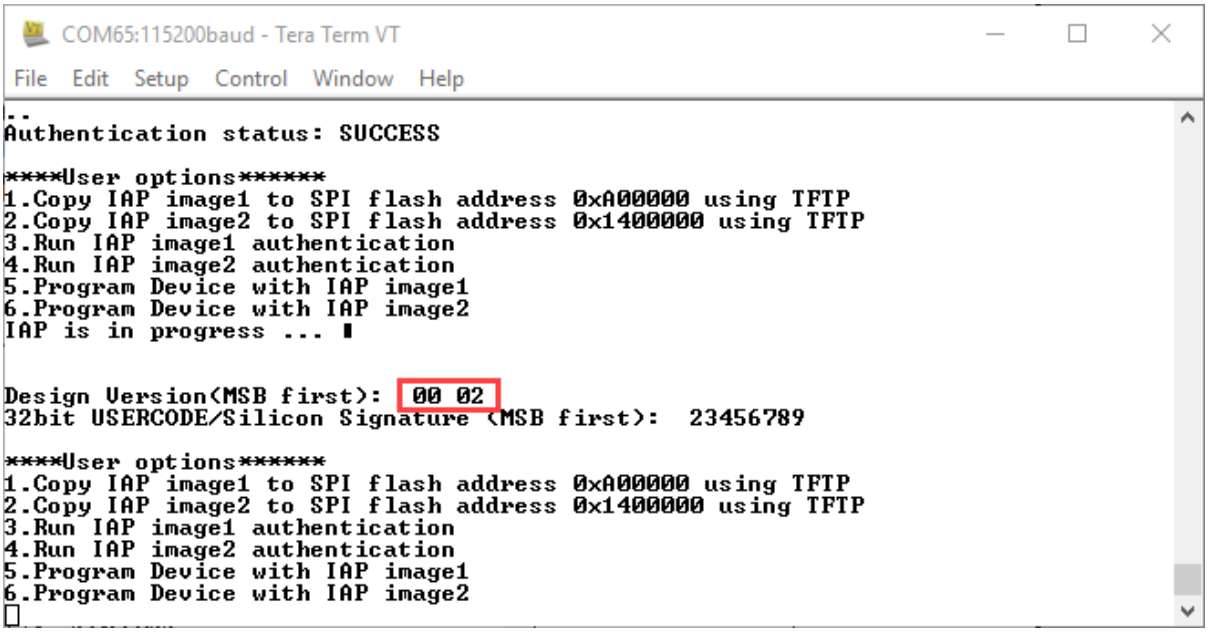
This concludes the IAP image authentication.

7.3.2 Running IAP Program (Ask a Question)

To run IAP with programming images, perform the following steps:

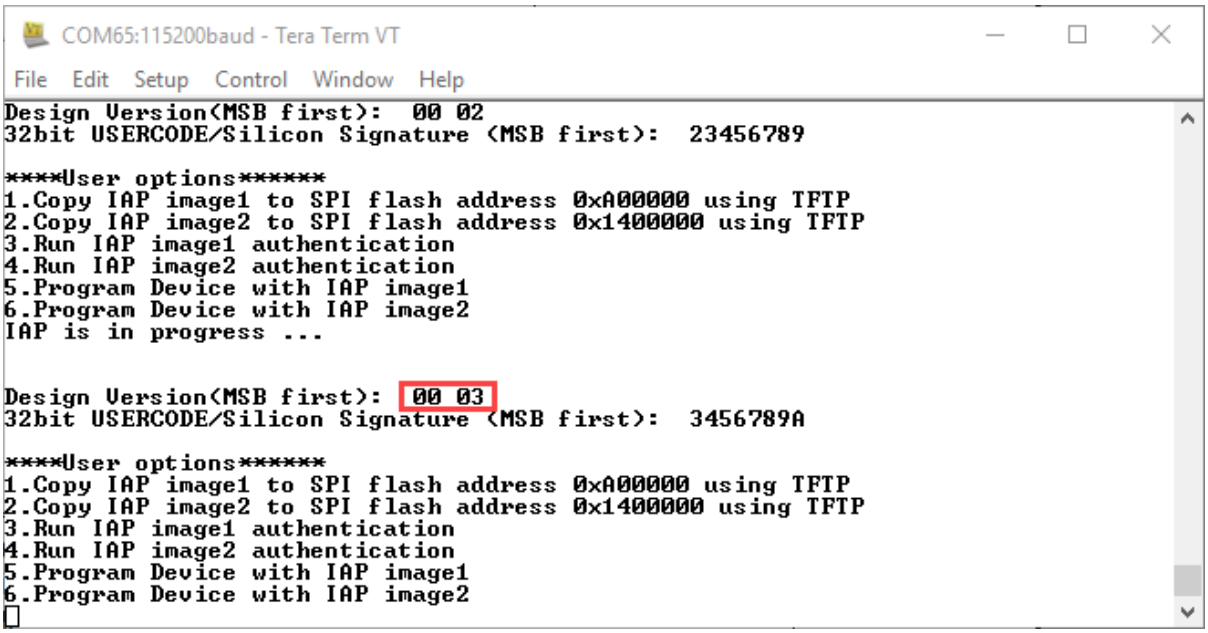
1. Press 5 to initiate **Program Device with IAP image1**. The IAP program with image1 is executed successfully and the design version **02** with different silicon signature is displayed as shown in the following figure. This operation takes few seconds.

Figure 7-17. Successful IAP with Image1



2. Press 6 to initiate **Program Device with IAP image2**. The IAP program with image2 is executed successfully and the design version **03** with different silicon signature is displayed as shown in the following figure. This operation takes a few seconds.

Figure 7-18. Successful IAP by Image2



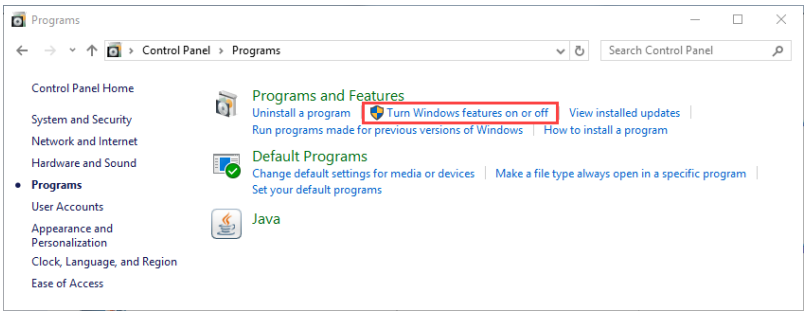
This concludes running the IAP Program with images.

8. Appendix 1: Enable TFTP Client (Ask a Question)

To enable TFTP client, follow these steps:

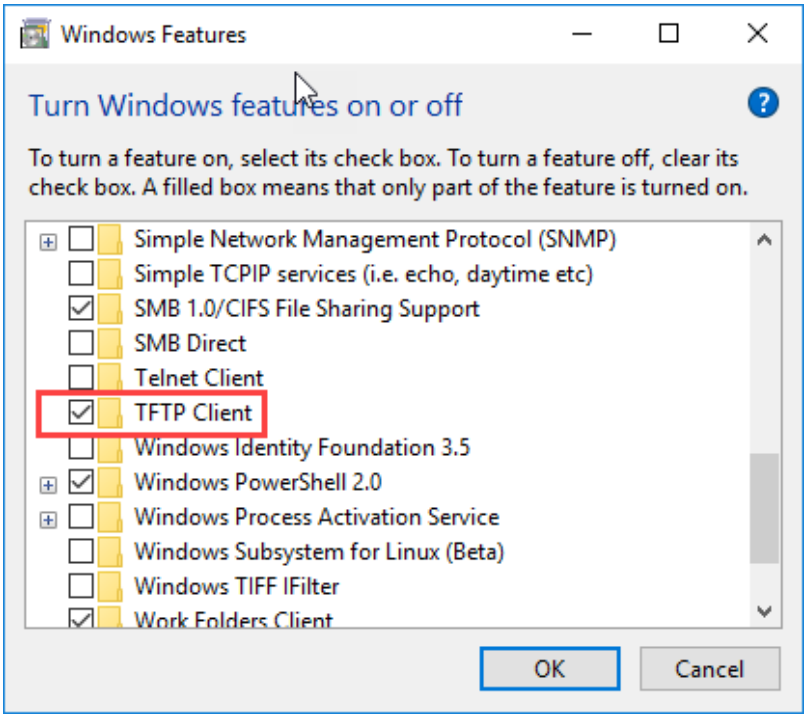
1. Navigate to **Control Panel** > **Programs**. Click **Turn Windows features on or off** as shown in the following figure.

Figure 8-1. Control Panel-Programs and Features



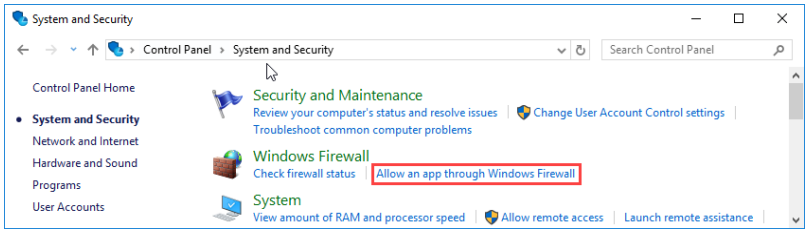
2. Select the **TFTP Client** check box from Windows® Features as shown in the following figure.

Figure 8-2. Selecting TFTP Client from Windows Features



3. Navigate to **Control Panel** >**System and Security** and click **Allow an app through Windows Firewall**.

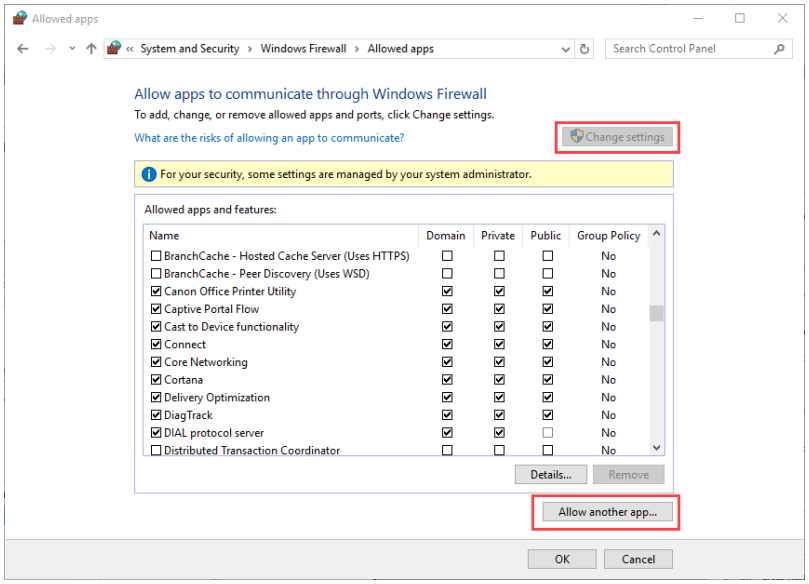
Figure 8-3. System and Security Window



Important: If the **System and Security** option is not available, then enter the firewall in the search window to perform step 3.

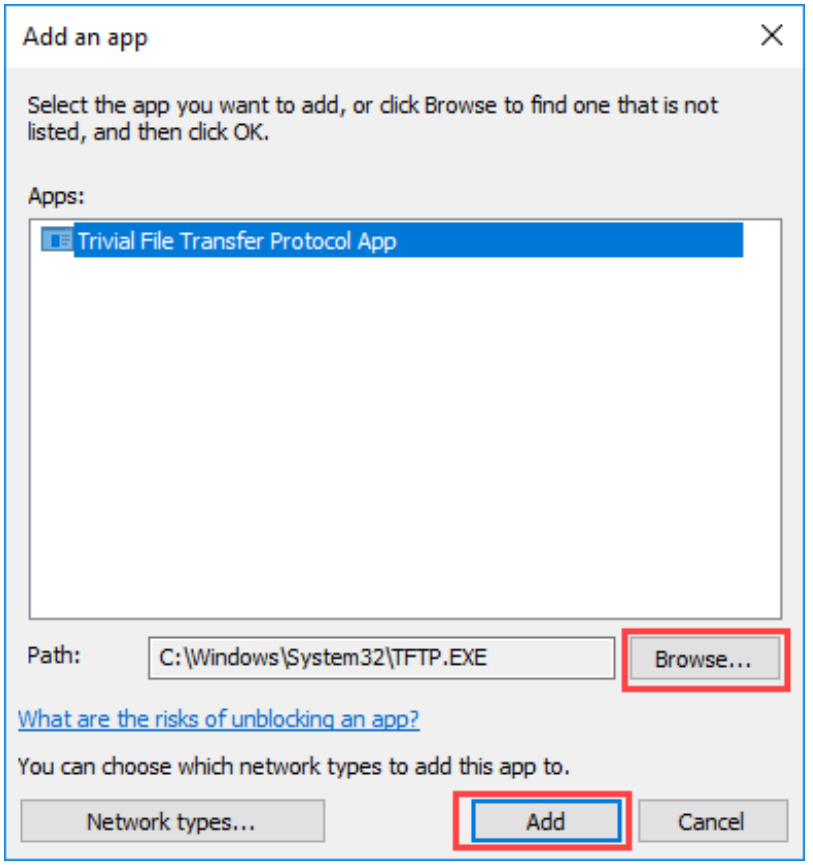
4. Click **Change Settings** and choose **Allow another program...**

Figure 8-4. Allow Programs Window



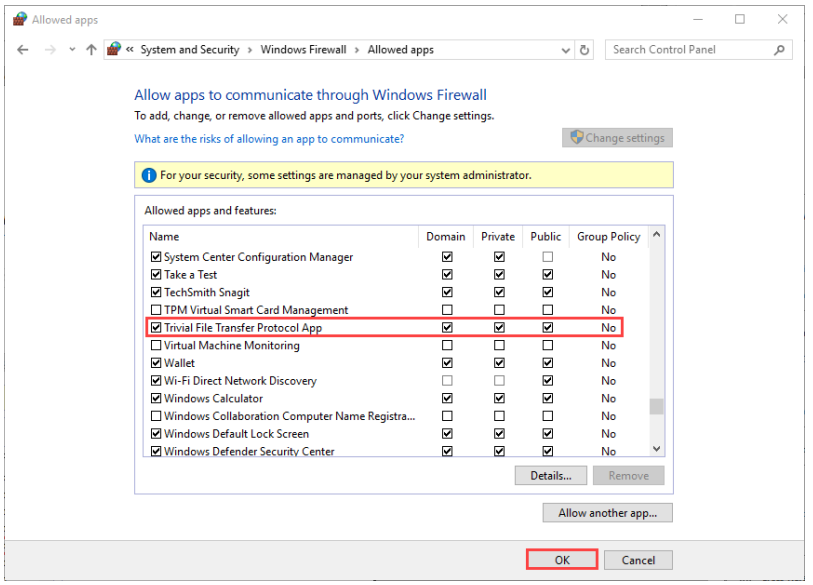
5. The **Add an app** window is displayed. Click **Browse...**
6. Navigate to **C**:\ -> **Windows**->**System32**. Choose **TFTP.exe** and click **Open**.
7. Ensure the TFTP.EXE path `C:\Windows\System32\TFTP.EXE` is selected correctly and click **Add**.

Figure 8-5. Add an app Window



8. Ensure the **Trivial File Transfer Protocol App** is added and select the check boxes (Domain, Home/Work, and Public) as shown in the following figure.

Figure 8-6. Selecting Trivial File Transfer Protocol App in Allowed apps and Features Window



9. Click **OK**.

9. Appendix 2: Running the SoftConsole Project in Debug Mode from LSRAM or DDR Memory (Ask a Question)

The following steps describe how to run the SoftConsole project in Debug mode.

1. Open the webserver or TFTP_ IAP application project using SoftConsole.
   - Webserver SoftConsole project location:

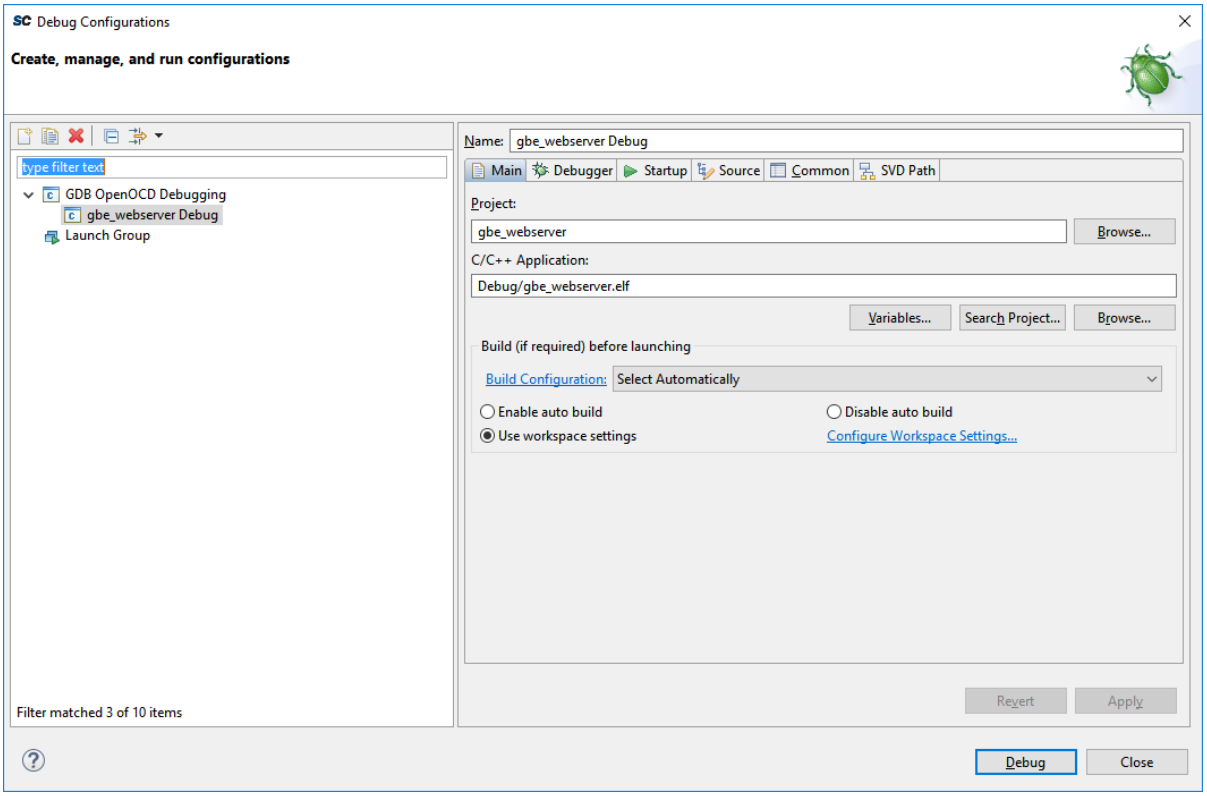
     ```
     mpf_an4569_v2023p2_eval_df\webserver\SoftConsole_Project
     ```

   - TFTP_ IAP SoftConsole project location:

     ```
     mpf_an4569_v2023p2_eval_df\tftp_iap\SoftConsole_Project
     ```

2. In SoftConsole, select **Run** > **Debug Configurations**. The **Debug Configurations** dialog box appears. To debug the webserver project, select **gbe_webserver Debug**, as shown in the following figure.

Figure 9-1. SoftConsole Debug Configuration

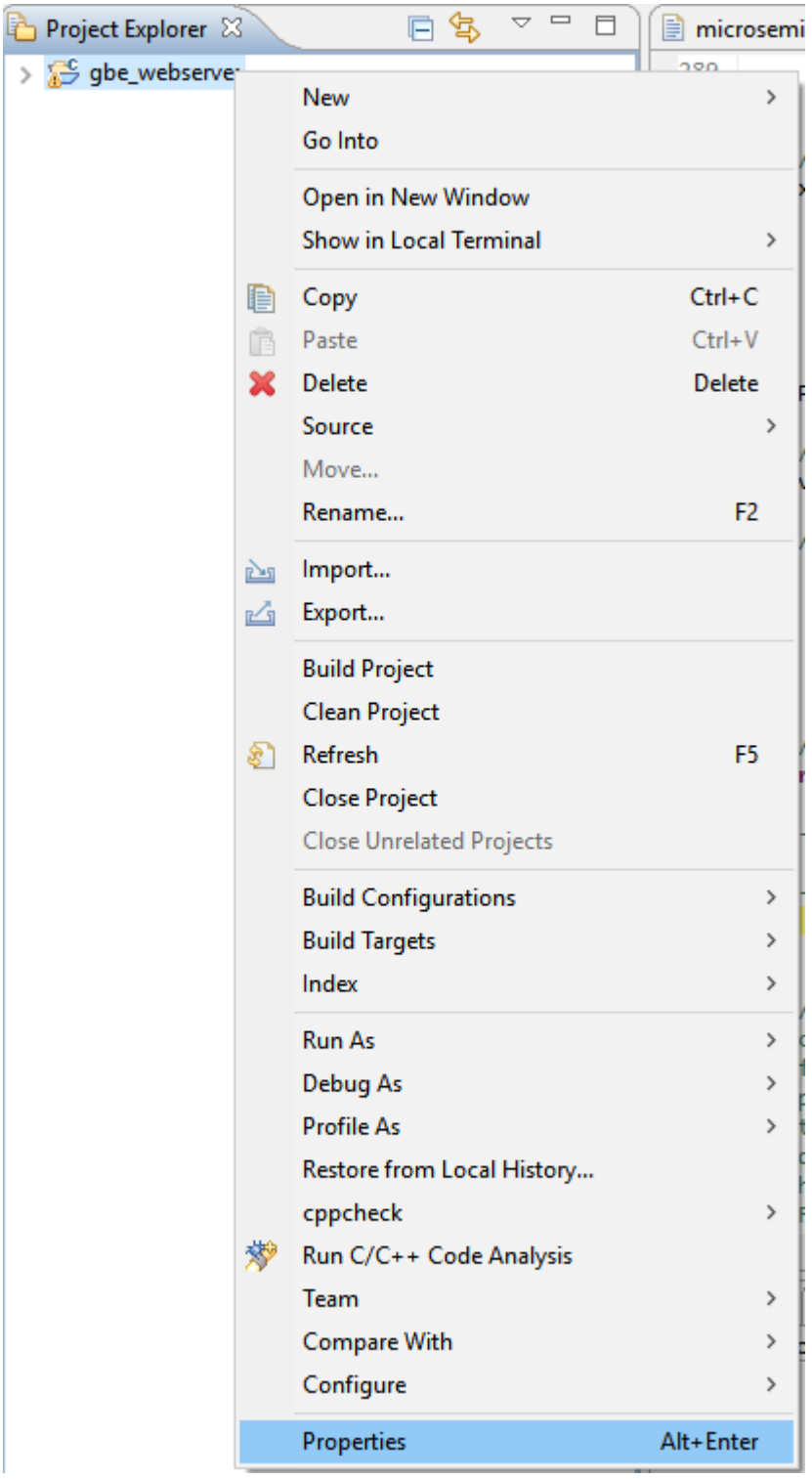


3. Click **Debug**. The tool copies the code to LSRAM memory and launches the debug session. This SoftConsole project is configured to debug from LSRAM.

To debug the application from the DDR memory, follow these steps:

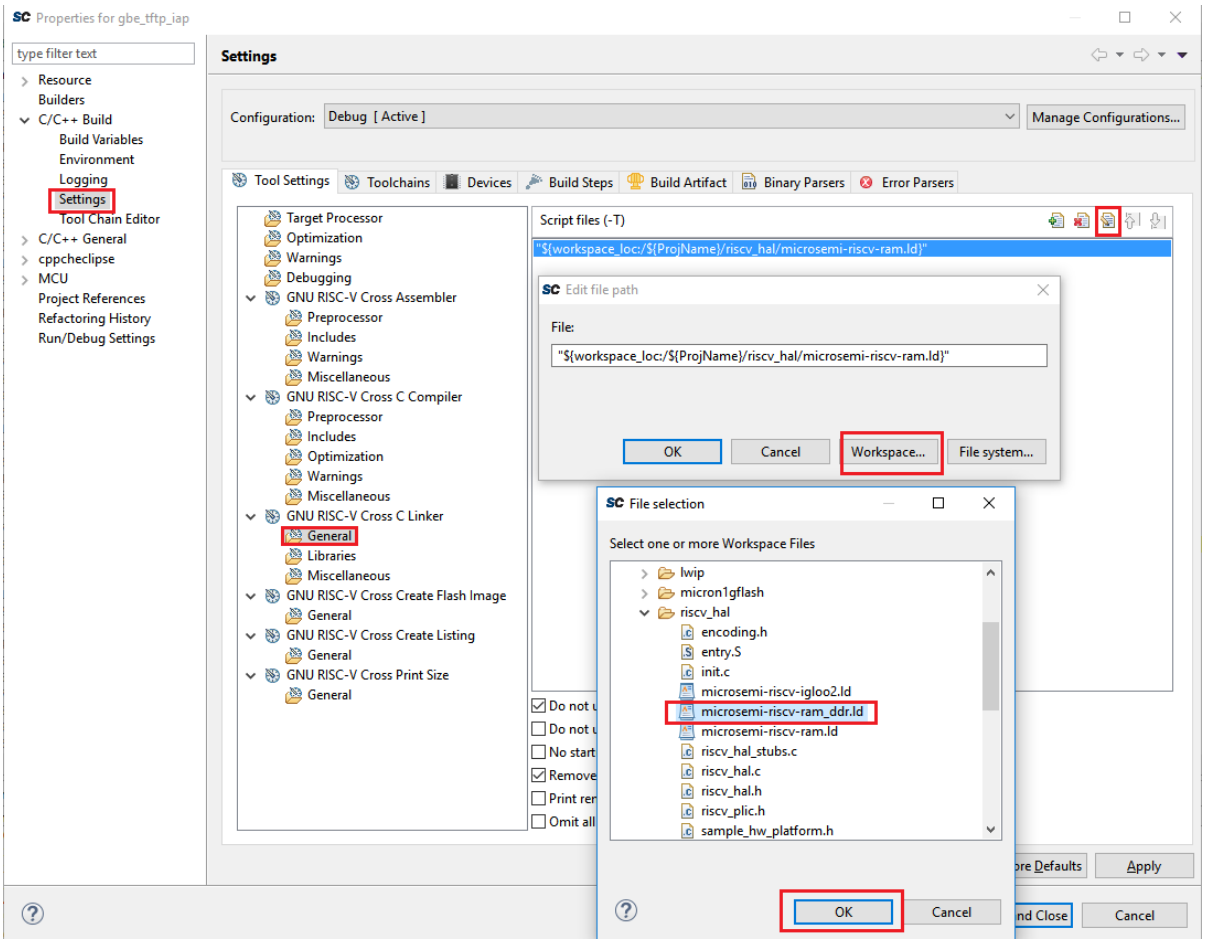
1. In the SoftConsole **Project Explorer** window, right click on **gbe_webserver project**, and select **Properties**, as shown in the following figure.

Figure 9-2. Project Explorer Window



2. Change the linker script file setting to `miv-rv32-ram-ddr.ld` and re-build the project, as shown in the following figure.

Figure 9-3. Project Properties



3. In SoftConsole, select **Run** > **Debug Configurations** to debug the application from DDR memory.

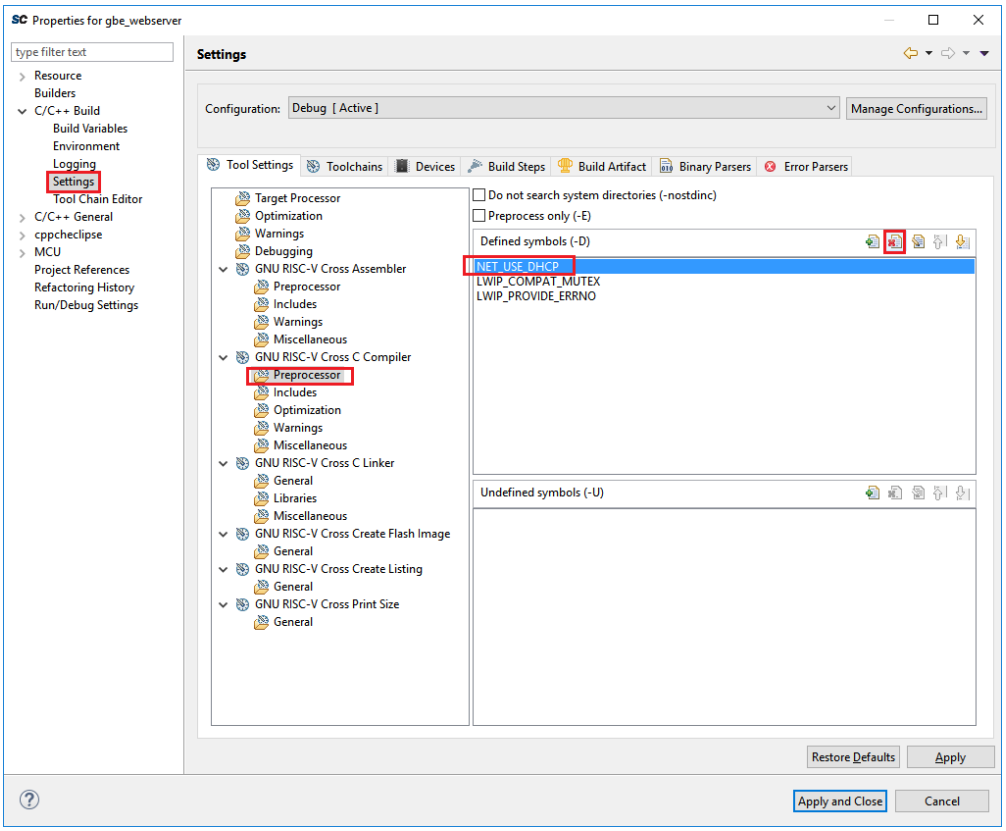
9.1 Running the Design in Static IP Mode (Ask a Question)

The following steps describe how to run the design in static IP mode.

Note: This procedure provide steps to run the webserver design. To run IAP using the TFTP design, perform the same steps by opening the IAP_TFTP project in SoftConsole.

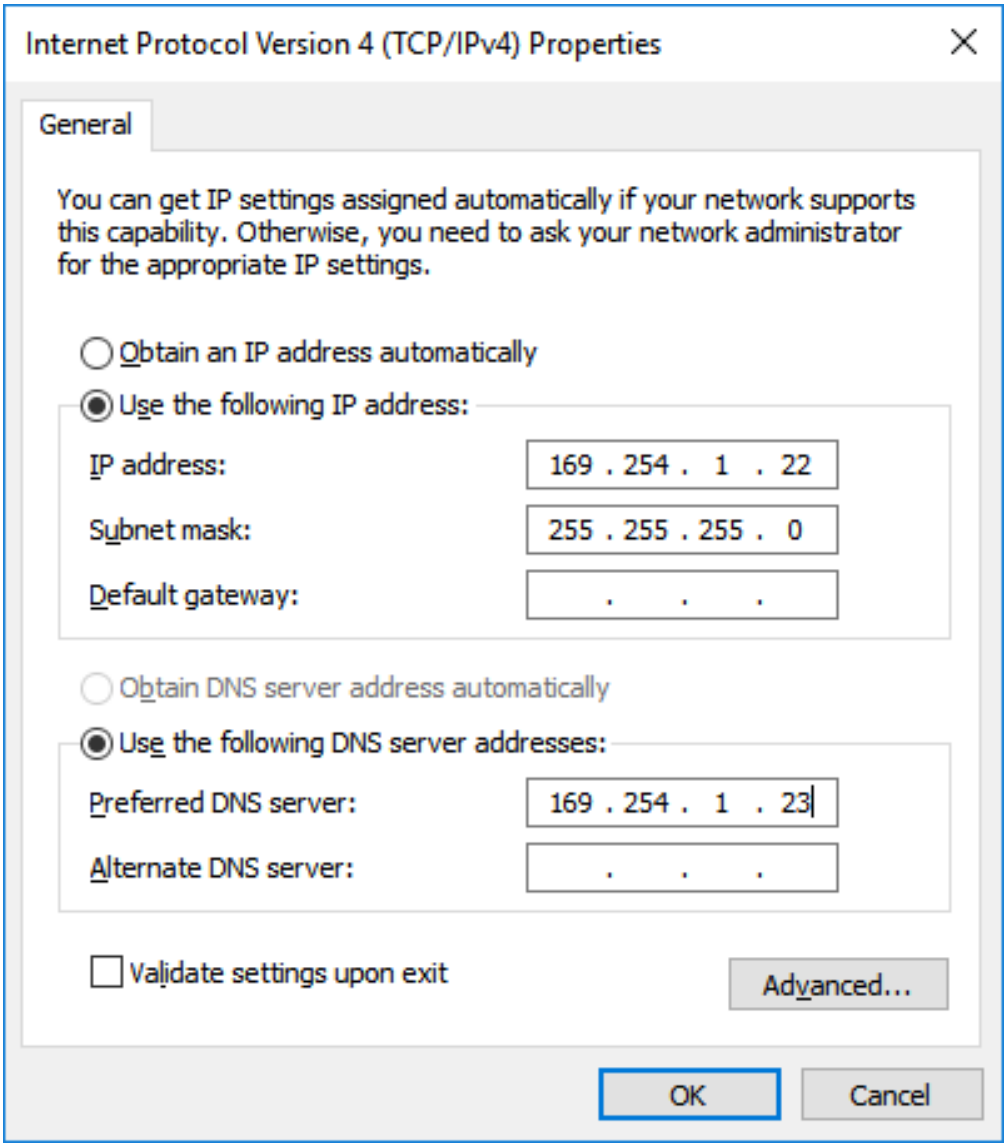
1. In SoftConsole **Project Explorer** window, right click the webserver (**gbe_webserver**) project, and select **Properties**, as shown in Figure 9-2.
2. In **Properties for gbe_webserver** window, remove the **NET_USE_DCHP** symbol listed under **Defined symbols (-D)**, and click **Apply and Close**, as shown in the following figure.

Figure 9-4. Properties for gbe_webserver



3. Change the host TCP/IP settings to connect with the board that has static IP address, **169.254.1.23**. The following figure shows the host PC TCP/IP settings.

Figure 9-5. Host PC TCP/IP Settings



4. Connect the PolarFire Evaluation board port J15 to Host PC using the RJ45 Ethernet cable.
5. After configuring the settings, compile the design, load it into the memory, and run it using SoftConsole. The Serial terminal shows board static IP:

```
/*CoreTSE WebServer using Mi-V SoftProcessor*/
Acquiring IP address.
169.254.1.23
```

6. Use the IP address in the web browser to display the Microchip web page.

10. Appendix 3: Programming the Device and External SPI FlashPro Express (Ask a Question)

This section describes how to program the PolarFire device and external SPI Flash with the .job programming file using FlashPro Express. The `.job` file is available at the following design files folder location:

- **Webserver**: `mpf_an4569_v2023p2_eval_df\webserver\programming_job`
- **TFTP_IAP**: `mpf_an4569_v2023p2_eval_df\tftp_iap\programming_job`

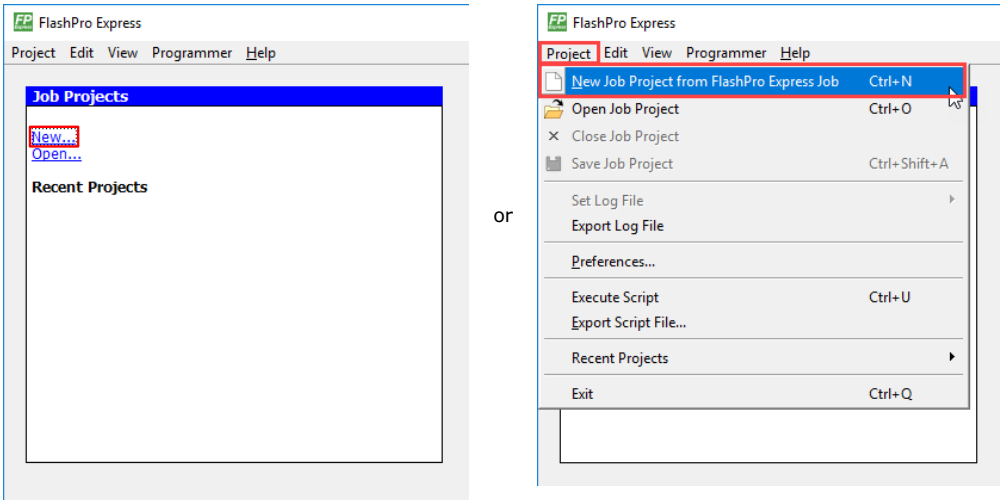
To program the device, perform the following steps:

1. Ensure that the jumper settings on the board are the same as listed in Table 6-2.

> **Important:** 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.

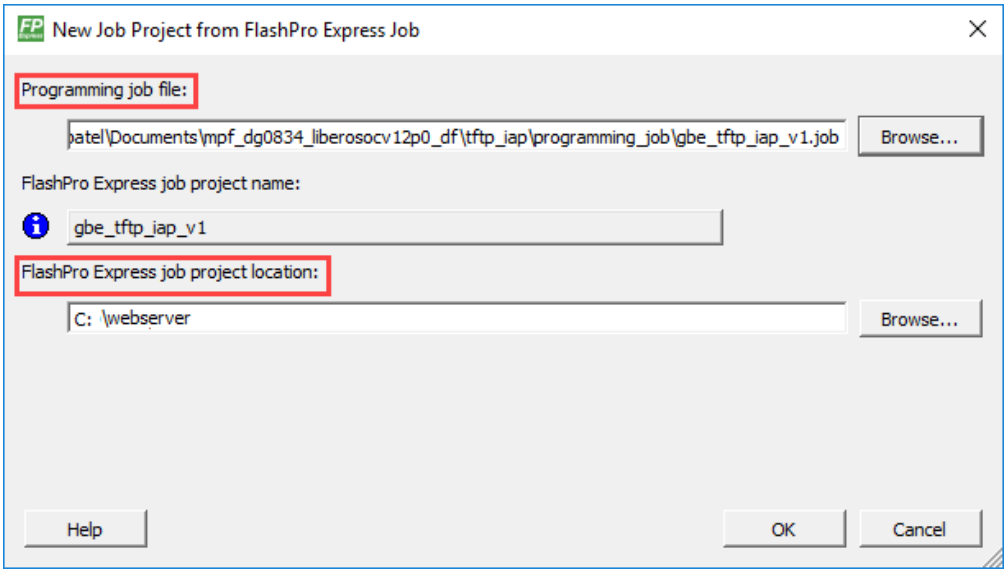
Figure 10-1. FlashPro Express Job Project



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:

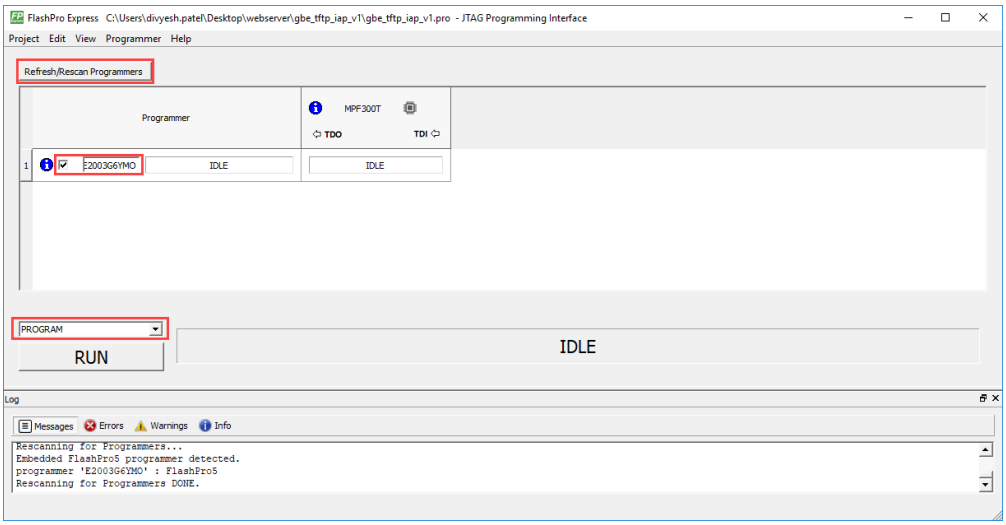
     `<download_folder>\mpf_an4569_v2023p2_eval_df\webserver\programming_job`
     or
     `<download_folder>\mpf_an4569_v2023p2_eval_df\tftp_iap\programming_job`.
   - **FlashPro Express job project location**: Click **Browse...** and navigate to the location where you want to save the project.

Figure 10-2. New Job Project from FlashPro Express Job



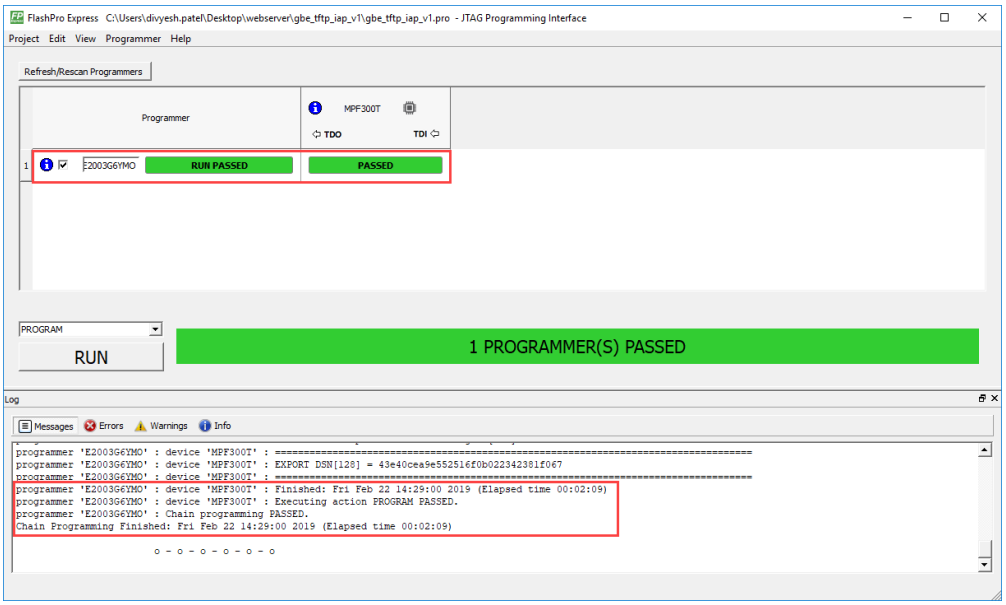
8. Click **OK**. The required programming file is selected and ready to be programmed in the device.
9. The FlashPro Express window appears as shown in the following figure. Confirm that a programmer number appears in the **Programmer** field. If it does not, check the board connections and click **Refresh/Rescan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

Figure 10-3. Programming the Device



10. Click **RUN**. When the device is programmed successfully, a **RUN PASSED** status is displayed as shown in the following figure. See 7. Running the Demo to run the webserver and the TFTP_IAP demo.

Figure 10-4. FlashPro Express-RUN PASSED



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

11. Appendix 4: Running the TCL Script (Ask a Question)

TCL scripts are provided in the design files folder under the directory TCL_Scripts. If required, the design flow can be reproduced from design implementation before the job file is generated.

To run the TCL script, follow these steps:

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, the Libero project is created within the TCL_Scripts directory.

For more information about TCL scripts, see Webserver: **mpf_an4569_v2023p2_eval_df/webserver/TCL_Scripts/readme.txt**. TFTP_IAP: **mpf_an4569_v2023p2_eval_df/tftp_iap/TCL_Scripts/readme.txt**.

See Tcl Commands Reference Guide for more details on TCL commands. Contact Technical Support for any queries encountered when running the TCL script.

12. Revision History (Ask a Question)

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

Revision	Date	Description
D	02/2024	Added a note in the 2.1.4. DDR3 section.
C	10/2023	The following is the list of changes in revision C of the document: • Updated Figure 2-2, Figure 2-3, Figure 2-4, Figure 2-5, Figure 2-6, Figure 2-8, Figure 2-9, Figure 2-10, and Figure 2-11. • Removed Design Memory Map section. • Updated directory file location in 5. SoftConsole Firmware Project. • Updated Figure 5-1 and Figure 5-2. • Updated directory file location in 6.5. Configure Design Initialization Data and Memories. • Updated directory file location in 9. Appendix 2: Running the SoftConsole Project in Debug Mode from LSRAM or DDR Memory. • Updated directory file location in 10. Appendix 3: Programming the Device and External SPI FlashPro Express. • Updated Tcl_scripts directory location in 11. Appendix 4: Running the TCL Script.
B	06/2022	The following is the list of changes in revision B of the document: • Removed hyperlink in 1.1. Application Layer. • Updated demo design files link in 4. Demo Prerequisites. • Updated directory file location in 5. SoftConsole Firmware Project. • Updated webserver hex file location in 6.5. Configure Design Initialization Data and Memories. • Updated browse folder location in 7.3. Running TFTP Demo. • Updated SoftConsole project location in 9. Appendix 2: Running the SoftConsole Project in Debug Mode from LSRAM or DDR Memory. • Updated `.job` file location in 10. Appendix 3: Programming the Device and External SPI FlashPro Express. • Updated Tcl_scripts directory location in 11. Appendix 4: Running the TCL Script.
A	06/2022	The following is the list of changes in revision A of the document: • The document was migrated to the Microchip template. • The document number was updated to DS00004569A from 50200834. • Updated the Table 3-1. • Updated the section 2. Demo Design. • Updated the Figure 2-2. • Updated the section 2.1.1. Mi-V Soft Processor. • Updated the section 2.1.2. PF_SRAM_AHBL_AXI. • Updated the Figure 2-7, Figure 2-8, Figure 2-9, and Figure 2-10. • Updated the Figure 2-11. • Updated the section 2.1.5. PF_CCC_0. • Updated the section 2.1.6. PF_IOD_CDR_CCC. • Updated the section 2.1.7. CORESPI_0. • Updated the section 2.1.11. CoreAPB3_0. • Updated the section 2.2. Clocking Structure. • Updated the section 6.5. Configure Design Initialization Data and Memories. • Updated the section 9. Appendix 2: Running the SoftConsole Project in Debug Mode from LSRAM or DDR Memory.

.....continued		
Revision	**Date**	**Description**
4.0	—	Added 11. Appendix 4: Running the TCL Script.
3.0	—	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.
2.0	—	The document was updated for Libero SoC v12.0 release.
1.0	—	Initial Revision.

Microchip FPGA Support

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

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

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

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

Microchip Information

The Microchip Website

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

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

Product Change Notification Service

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

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

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

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

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

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

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

Legal Notice

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

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

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

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

Trademarks

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

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

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic

ISBN: 978-1-6683-4086-8

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

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