AN4568

PolarFire® SmartSFP+ Solution Featuring In-Application Programming

Introduction

Microchip's SmartSFP+ is a system-level optical transceiver solution built with the smallest form factor and the lowest power PolarFire FPGA. The SmartSFP+ module combines the flexibility and advantage of data processing and complex data handling operations in a single FPGA device within the module. It saves processing overhead in communication protocols as the data is processed by upper layers running within the module. The SmartSFP+ module supports in-application programming (IAP) which retains its flexibility even after its deployment in end products.

The SmartSFP+ module supports the following key features:

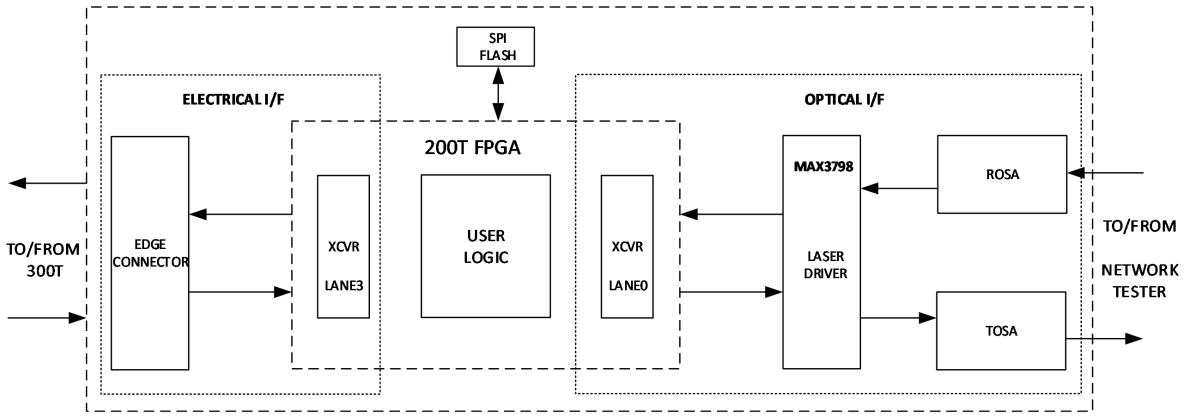
- Supports an LC-type connector.
- Supports multi-mode optical fiber.
- Supports data rates upto 10 Gbps.
- Works with 850 nm and 1310 nm wavelengths.
- Supports FTTH, CPRI, Fiber Channel, Ethernet (10GBASE-R) with SyncE capability.

The main components of the SmartSFP+ module are as follows:

- PolarFire 200T FPGA with two SerDes lanes supporting data rates upto 12.7 Gbps.
- Optical interface comprising of the ROSA and TOSA modules which can support data rates upto 10 Gbps.
- MAX 3798 Laser Driver and Limiting Amplifier.
- 32 MB SPI Flash.
- Edge connector which is a part of the electrical interface.

The following figure shows the high-level block diagram of the PolarFire SmartSFP+ module.

Figure 1. Block Diagram of the SmartSFP+ Module



This document highlights a prototype solution, in which, the SmartSFP+ module interfaces with the PolarFire Evaluation Kit to demonstrate the following capabilities:

- Data loopback within the SmartSFP+ module through its optical interface.
- Data loopback from the SmartSFP+ module to the PolarFire Evaluation Kit through the electrical interface.
- IAP on the SmartSFP+ module initiated by the PolarFire Evaluation Kit.

Table of Contents

Introduction....1

1. Design Description....3
   - 1.1. IAP Initiator....3
   - 1.2. IAP Target....17
   - 1.3. Clocking Structure....23
   - 1.4. Reset Structure....24

2. SPI Programming Images....25
   - 2.1. Programming Options for GOLDEN Image....25
   - 2.2. Exporting the GOLDEN Image....25
   - 2.3. Programming Options for IAP Image....26
   - 2.4. Exporting the IAP Image....27

3. Demo Requirements....29

4. Demo Prerequisites....30

5. Setting Up the Demo....31
   - 5.1. Setting Up the Hardware....31
   - 5.2. Programming the MPF300T Device....32
   - 5.3. Programming the MPF200T Device....33
   - 5.4. Generating the IAP Update Design....33
   - 5.5. Installing the GUI....34

6. Running the Demo....35
   - 6.1. Executing IAP....35
   - 6.2. Executing Data Loopback....42

7. Revision History....45

Microchip FPGA Support....46

Microchip Information....46
   - The Microchip Website....46
   - Product Change Notification Service....46
   - Customer Support....46
   - Microchip Devices Code Protection Feature....46
   - Legal Notice....47
   - Trademarks....47
   - Quality Management System....48
   - Worldwide Sales and Service....49

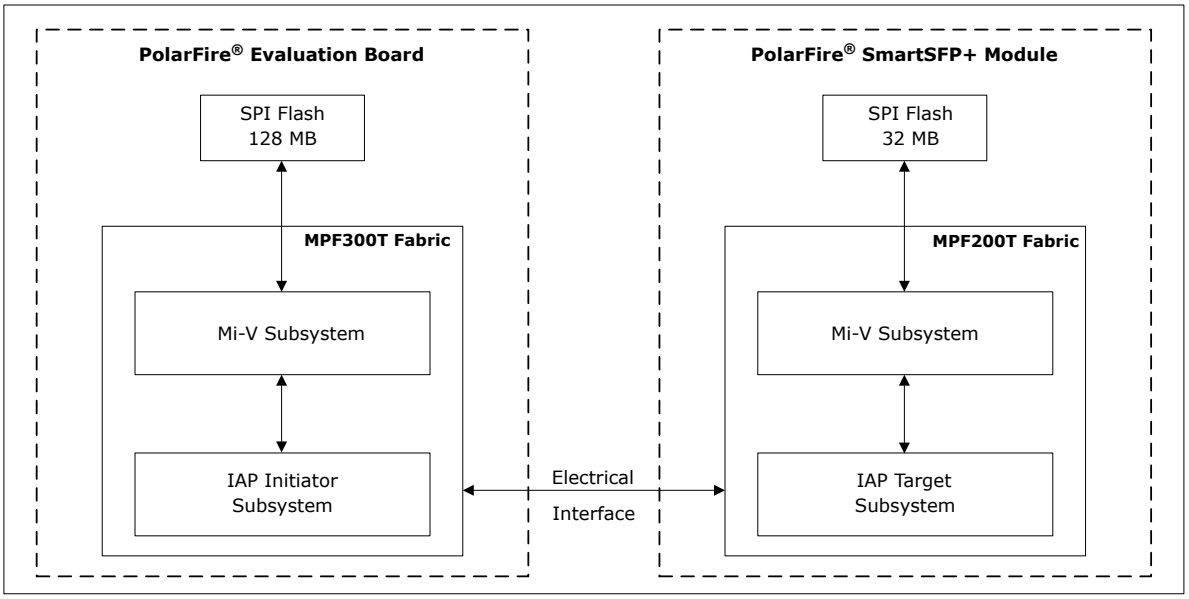
1. Design Description

The solution consists of the following blocks:

- **IAP Initiator**: IAP Initiator runs on the PolarFire Evaluation Kit and initiates the following functions:
  - Reads the `.spi` bitstream from the on-board 128 MB SPI Flash and transfers the bitstream in the form of Ethernet packets each with 1 KB of data payload. This operation completes when the entire bitstream is transferred to the target.
  - Generates and sends the authenticate command to the target for authenticating the new bitstream.
  - Generates and sends the program command to the target subsystem for executing the IAP using the newly stored bitstream.
  - Gets the version number of the target device.
  - Switches the design to IAP or Loopback mode based on user's selection (Graphical User Interface).
- **IAP Target**: IAP Target runs on the SmartSFP+ module. It interacts with IAP Initiator and performs the following functions:
  - Receives commands from IAP Initiator.
  - Receives the IAP bitstream data and stores in the external 32 MB SPI Flash.
  - Authenticates the received IAP Image.
  - Executes the IAP sequence.

The following figure shows the IAP Initiator and IAP Target subsystems of the solution.

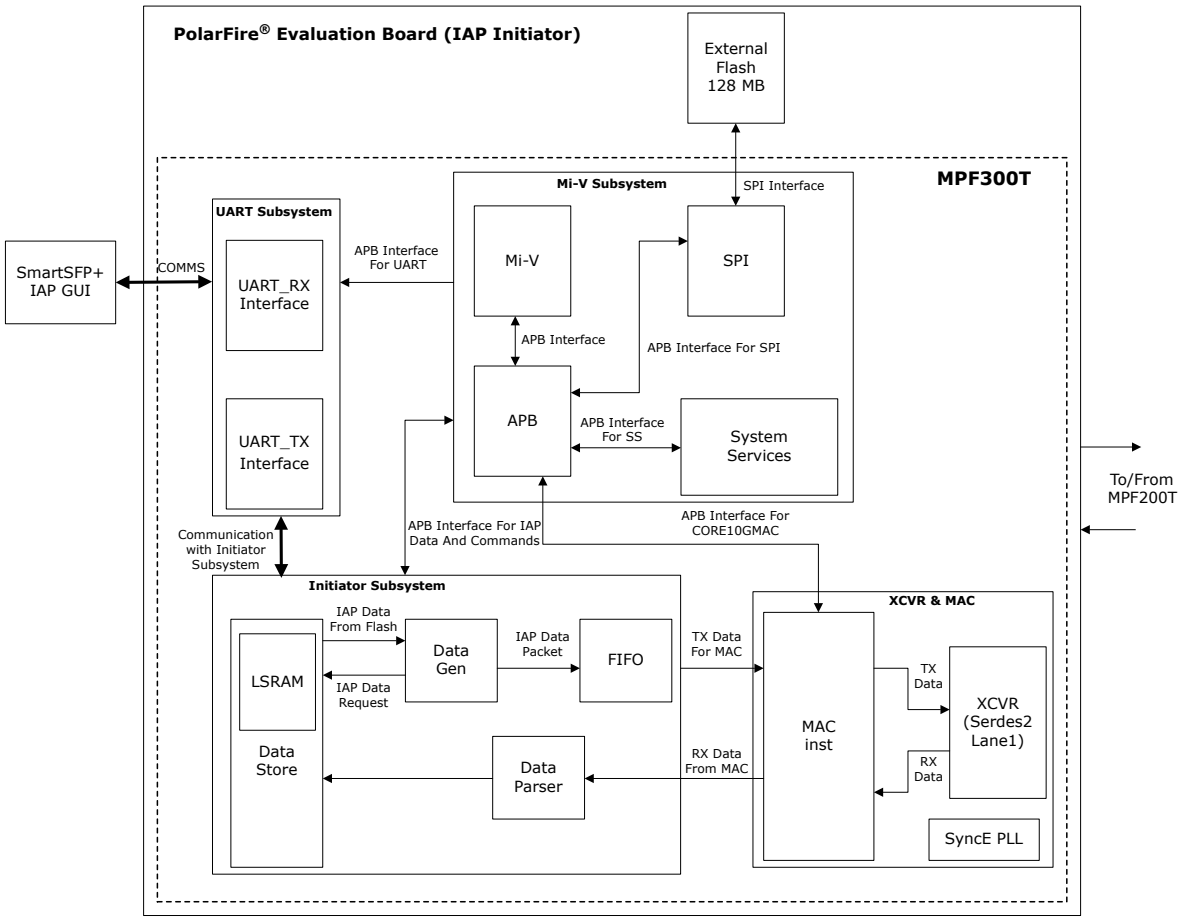
Figure 1-1. Top-Level Block Diagram



1.1 IAP Initiator

The following figure shows the functional blocks of IAP Initiator.

Figure 1-2. IAP Initiator Block Diagram



The main components of IAP Initiator are as follows:

- 1.1.1. Mi-V Subsystem (IAP Initiator)
- 1.1.2. UART Subsystem
- 1.1.3. Initiator Subsystem
- 1.1.4. PF_XCVR_0 (IAP Initiator)
- 1.1.5. CORE10GMAC0 (IAP Initiator)

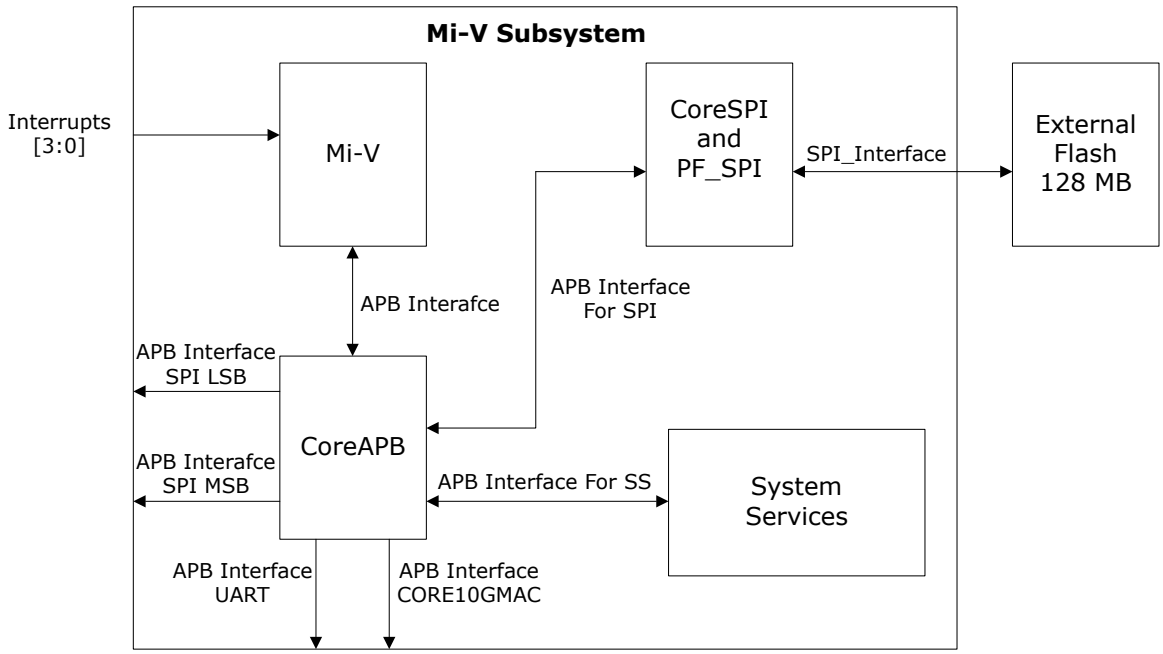
1.1.1 Mi-V Subsystem (IAP Initiator)

The Mi-V subsystem in IAP Initiator handles the following user commands from the GUI:

1. Get Target Version: To generate an interrupt for fetching the design version from the Target device (IAP Target).
2. Image Transfer: To transfer the bitstream from IAP Initiator to IAP Target in the form of Ethernet packet.
3. Authenticate IAP Image: To authenticate the stored IAP bitstream.
4. Program the Image: To execute the IAP sequence in IAP Target.
5. Clear the Sequence: To clear the status of executed commands by IAP Initiator and IAP Target.

The following figure shows sub-blocks of the Mi-V subsystem.

Figure 1-3. Mi-V Subsystem Sub-blocks



The Mi-V subsystem comprises of the following components:

- 1.1.1.1. Mi-V Soft Processor
- 1.1.1.2. CoreGPIO_0 (IAP Initiator)
- 1.1.1.3. CoreSPI_0 (IAP Initiator)
- 1.1.1.4. CoreAPB3_0 (IAP Initiator)

1.1.1.1 Mi-V Soft Processor

The Mi-V soft processor's default Reset Vector Address value is `0x8000_0000`. After the device Reset, the soft processor executes the application from TCM, which is mapped to `0x8000_0000`, hence the Reset Vector Address is set to `0x8000_0000` as shown in Figure 1-4. TCM is the main memory of the Mi-V soft processor. TCM gets initialized with the user application from sNVM. In the Mi-V soft processor's memory map, the `0x8000_0000 – 0x8000_FFFF` range is defined for the TCM memory interface and the `0x6000_0000 – 0x6FFF_FFFF` range is defined for the APB interface. The following figures show these configuration settings.

Figure 1-4. Mi-V TCM and Interface Configuration

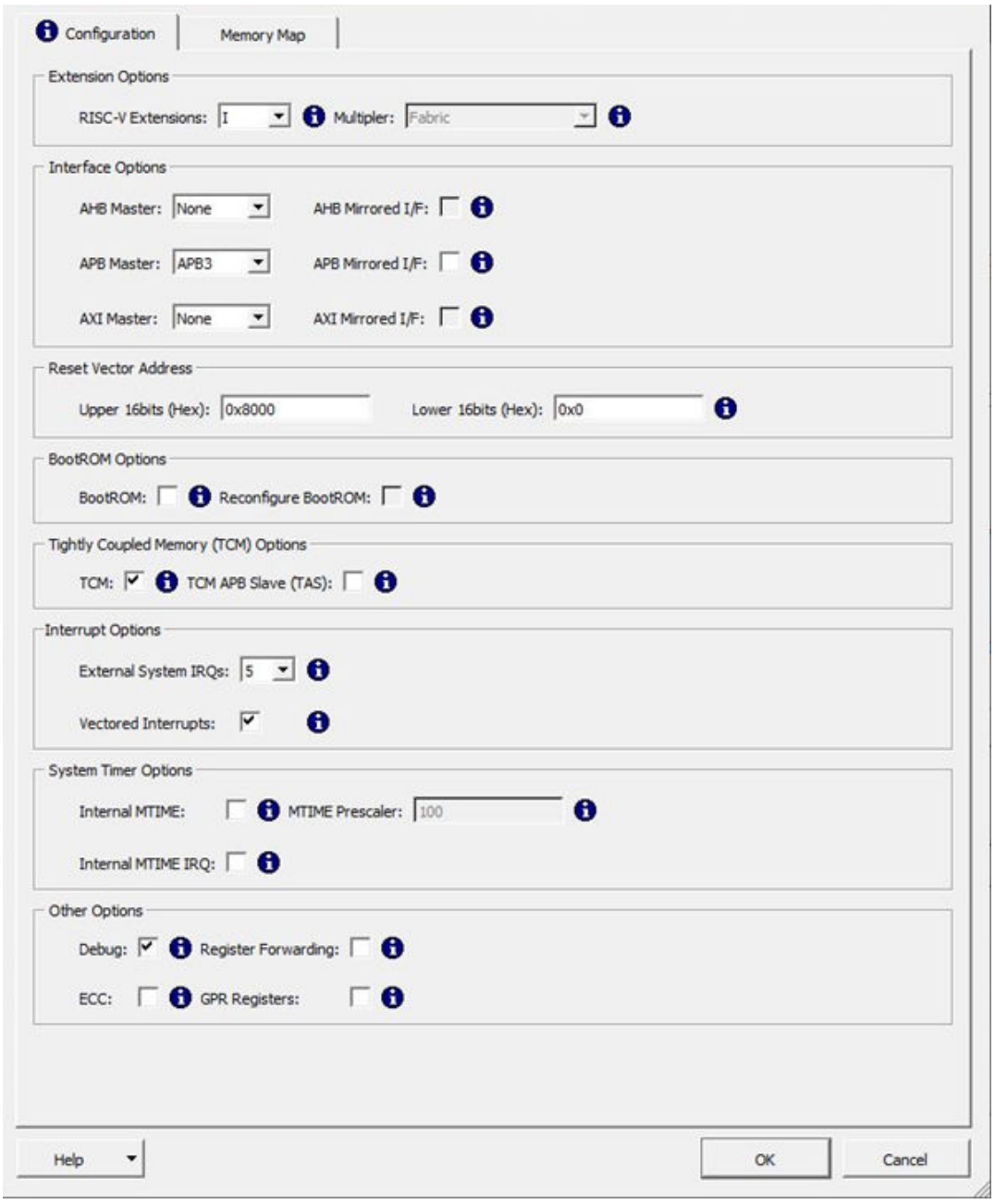
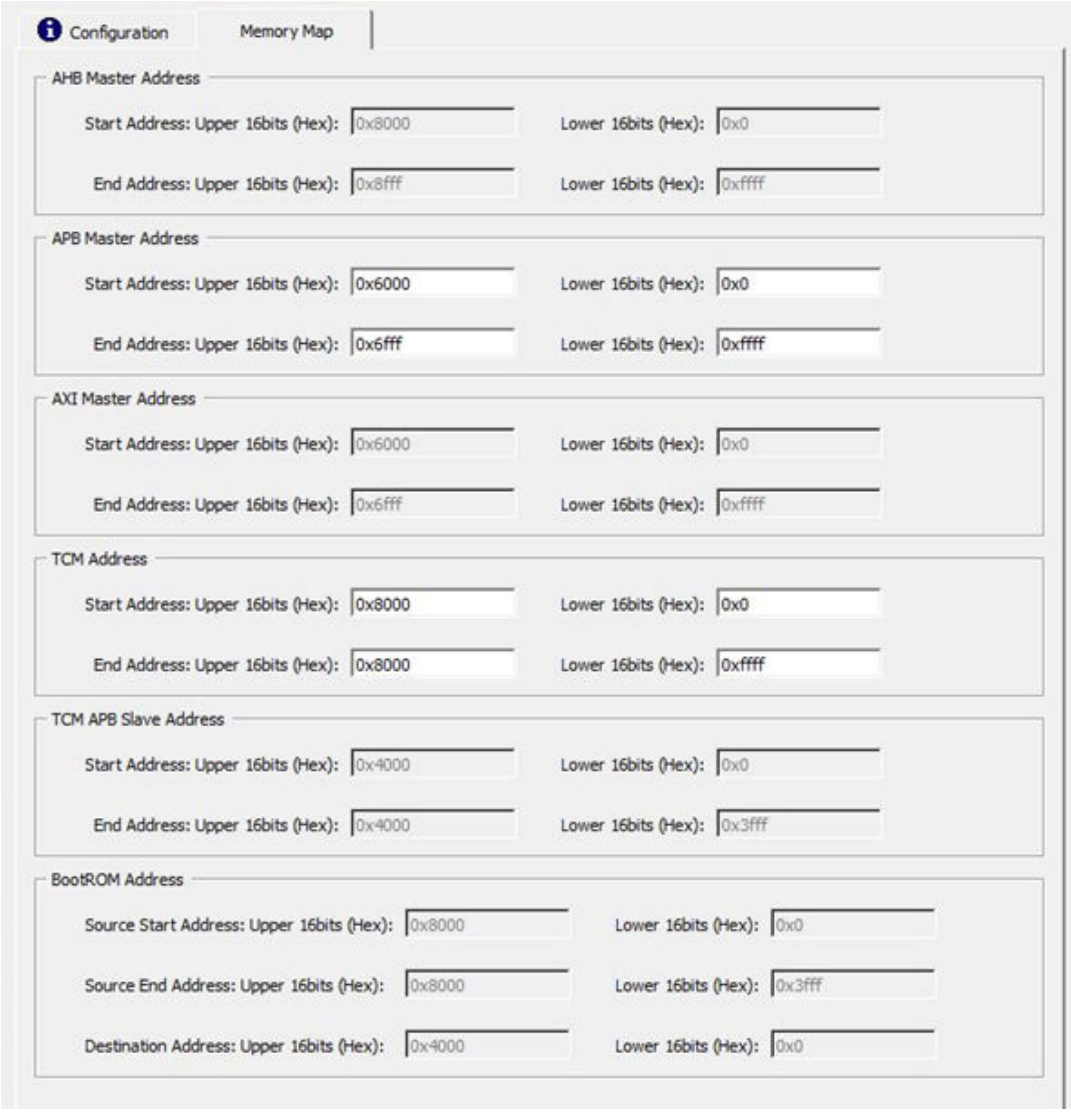


Figure 1-5. Mi-V APB Memory Map Configuration



Note: Memory depth of TCM is set to 16384 words to accommodate an application of up to 64 KB into it. The present version of the application is below 50 KB so this can fit into either sNVM or µPROM. In this design, sNVM is selected as the data storage client.

1.1.1.2 CoreGPIO_0 (IAP Initiator)

The CoreGPIO_0 IP controls the on-board LEDs using GPIOs. It is connected to the Mi-V soft processor as an APB slave. The configuration settings of CoreGPIO_0 are as follows:

- **Global Configuration**
  - **APB Data Width**: 32
    The design uses 32-bit data width for APB read and write data.
  - **Number of I/Os**: 1
    The design controls one on-board LEDs for output.
  - **Single-bit interrupt port**: Disabled
  - **Output enable**: External
- The following list shows the sub-options under **I/O Bit 0**.

- **Output on Reset**: 0
- **Fixed Config**: Yes
- **I/O Type**: Output
- **Interrupt Type**: Disabled
  When I/O states change, no interrupt is required for the application.

1.1.1.3 CoreSPI_0 (IAP Initiator)

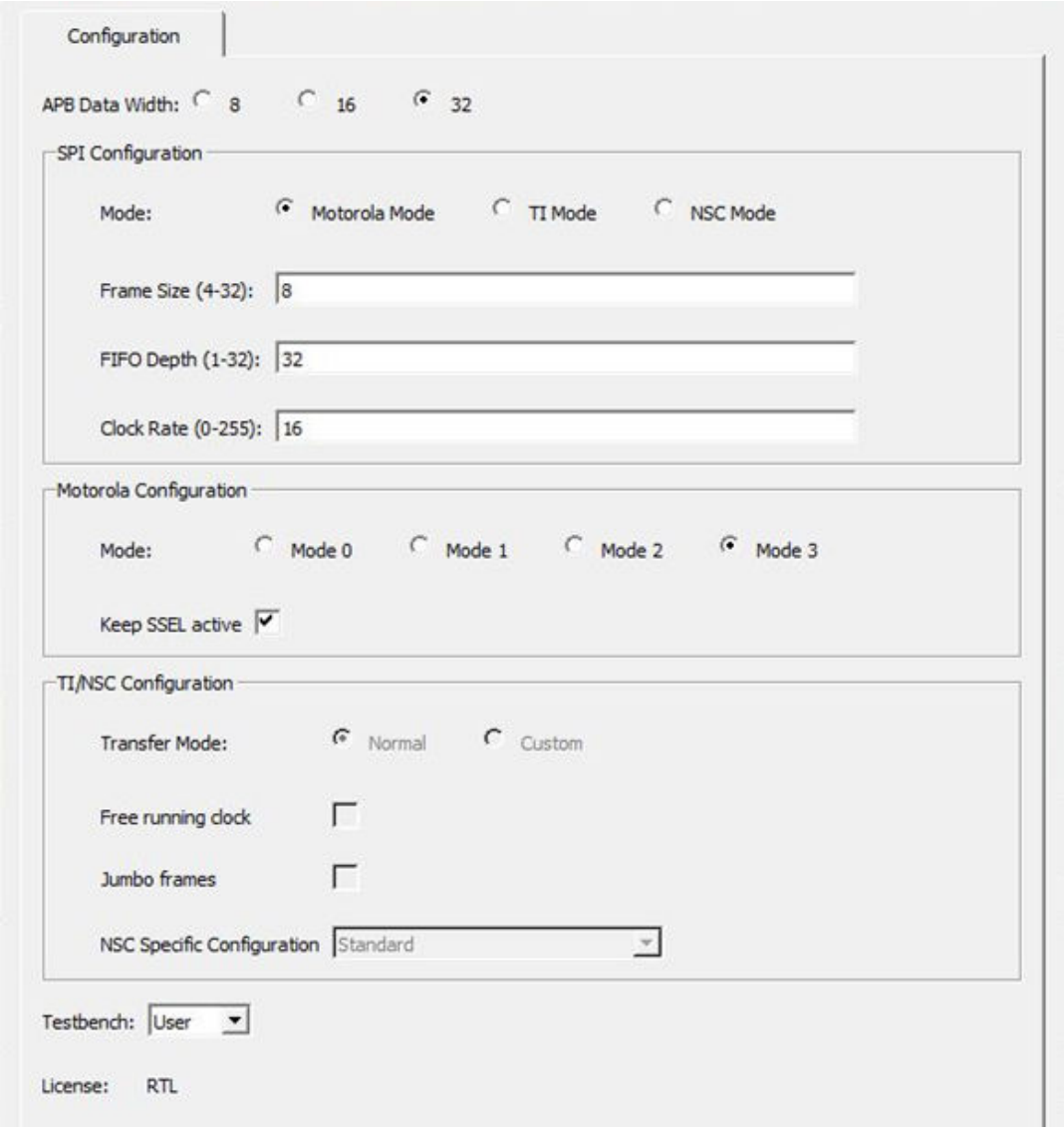
The CoreSPI IP is used to program the external SPI Flash using the Mi-V processor. This IP interfaces the fabric logic to the external SPI Flash, which is connected to the system controller.

- **APB Data Width**: Select 32 as APB data width in the design. The default value is 8.
- **SPI Configuration**
  - **Mode**: Motorola Mode (default) as the target SPI slave supports Motorola Mode. Mode 3 is selected under **Motorola configuration**.
  - **Frame Size**: 8
  - **FIFO Depth**: 32
    To store maximum frames(TX and RX) in FIFO.
  - **Clock Rate**: 16
- **Motorola Configuration**
  - **Mode**: Mode 3
  - **Keep SSEL Active**: Enabled
    To keep the slave peripheral active between back to back data transfers.

SPI Clock = System Clock/2x(16+1).

The following figure shows the CoreSPI_0 configuration.

Figure 1-6. CoreSPI Configuration

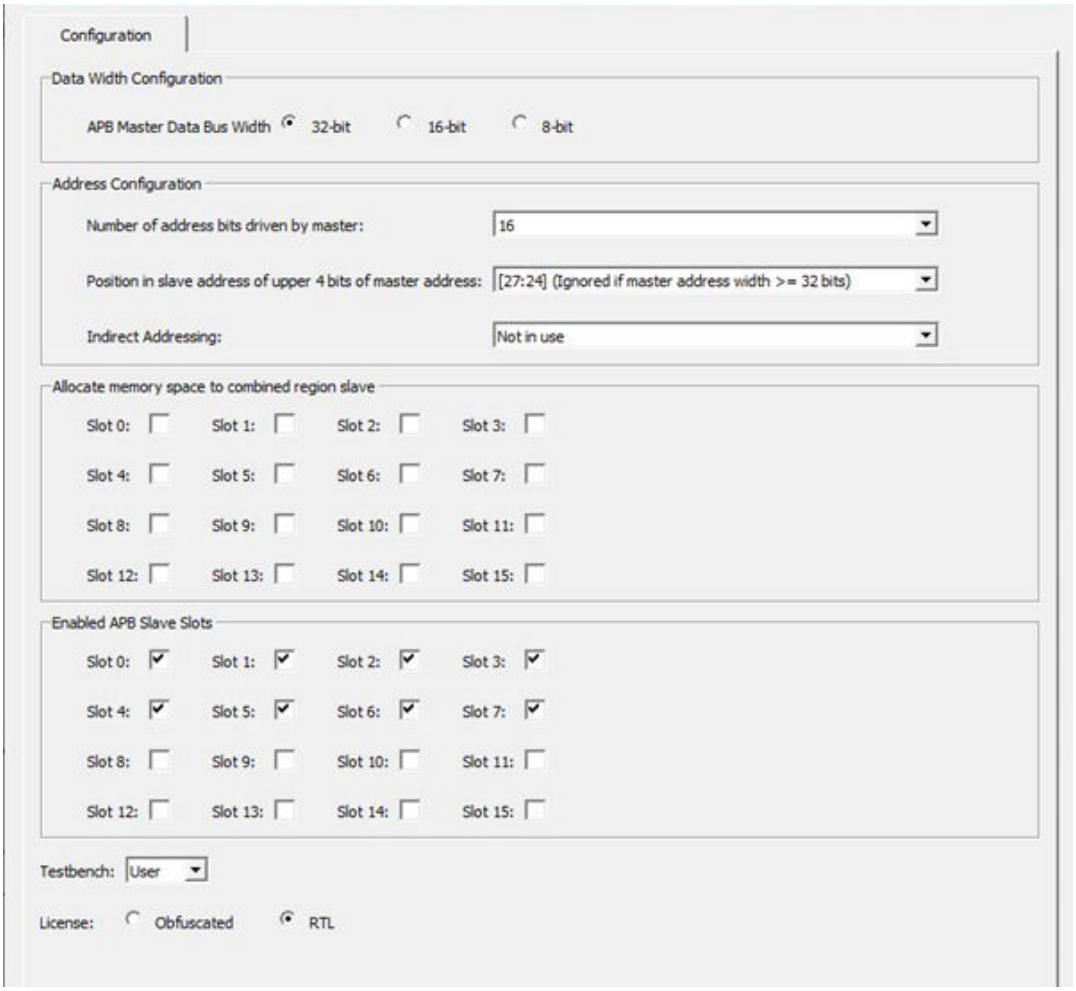


1.1.1.4 CoreAPB3_0 (IAP Initiator)

The CoreAPB3 IP connects the peripherals, PF_SYSTEM_SERVICES, CoreSPI, and CoreGPIO as slaves to the Mi-V soft processor. This IP is configured as follows:

- **APB Master Data Bus Width**: 32-bit
  The design uses 32-bit data width for APB read and write data.
- **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
- **Enabled APB Slave Slots**: Slot 0, Slot 1, Slot 2, Slot 3, Slot 4, Slot 5, Slot 6, and Slot 7.

Figure 1-7. CoreAPB3 Configuration (IAP Initiator)



CoreAPB3_0 block interacts with Mi-V and has the following slave interfaces:

- **APB IF UART:** (0x6000_0000 to 0x6000_0FFF).
  The APB interface is used by the firmware to get the number of transactions and the last burst size information from the user interface. A dedicated register is used to store the current packet number for the user interface.
- **APB_IF_CORE10GMAC:** 0x6000_6000 to 0x6000_6FFF
- **APB_IF_ LSBDATA:** 0x6000_4000 to 0x6000_4FFF
  Firmware writes the LSb part of the SPI image to the fabric.
- **APB_IF_ MSBDATA:** 0x6000_5000 to 0x6000_5FFF
  Firmware writes the MSb part of the SPI image to the fabric.
- **APB_IF_SS :** 0x6000_2000 to 0x6000_2FFF
  This APB interface sends commands to the Core system services.
- **APB_IF_SPI:** 0x6000_3000 to 0x6000_3FFF
  This APB interface sends commands and receives data from the external SPI Flash via system controller's PF_SPI and CoreSPI.

1.1.2 UART Subsystem

The UART subsystem provides a serial communication interface for passing user commands from GUI to IAP Initiator, and for displaying the status of the initiated commands. The UART subsystem has the following major components:

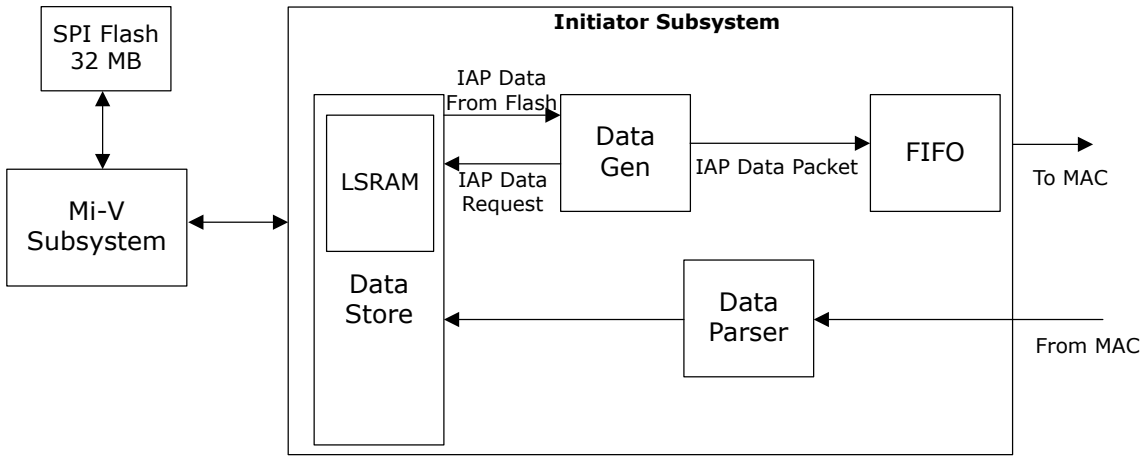
- Uart_rx_if : This finite state machine receives the write request and write data from COREUART_0 and generates the interrupts.
- Uart_tx_if: This finite state machine receives the following design status codes.
  - Image transfer complete
  - Authentication complete
  - Current packet number
  - Target version number

  These status codes are sent to COREUART_0 after receiving a read request.

1.1.3 Initiator Subsystem

The following figure shows the Initiator subsystem.

Figure 1-8. Initiator Subsystem



This block consists of the following sub-blocks:

- 1.1.3.1. Data Store
- 1.1.3.2. Data Gen
- 1.1.3.3. Data Parser

1.1.3.1 Data Store

The Data Store module is the access point to Mi-V for executing commands and for reading or writing the status of executed commands. This module generates a data read request to the Mi-V soft processor and stores the data in a TPSRAM (Two-Port SRAM). The data is read by the Data Gen module and transmitted to IAP Target.

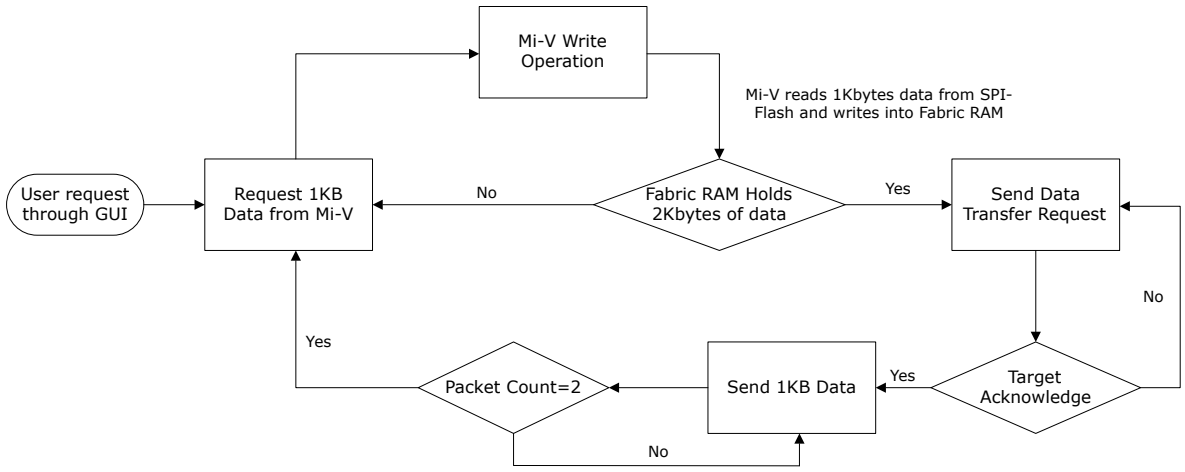
The Data Store module handles the following user requests:

- 1.1.3.1.1. Image Transfer
- 1.1.3.1.2. Get Target Version
- 1.1.3.1.3. Start IAP Authentication
- 1.1.3.1.4. Start IAP Program

1.1.3.1.1 Image Transfer

The following figure shows the `Image Transfer` command execution.

Figure 1-9. `Image Transfer` Sequence

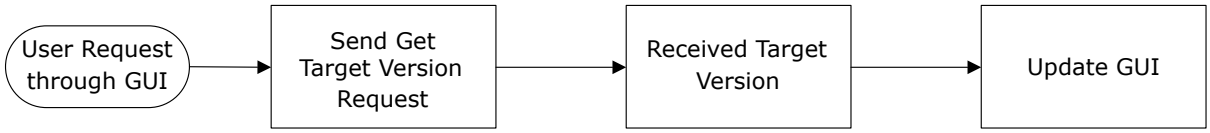


1. When the `Image Transfer` command is received from the user, the Mi-V soft processor reads 2 KB of data from SPI Flash and stores in the TPSRAM.
2. The Data Store module sends a `Image Transfer` request to IAP Target in the form of a control packet.
3. After receiving the acknowledgment from IAP Target, the Data Store module issues a data ready instruction to the Data Gen module. Data Gen reads the data from the TPSRAM and transmits the whole 2 KB data in bursts of two 1 KB successions.

1.1.3.1.2 Get Target Version

The following figure shows the sequence of `Get Target Version` command execution.

Figure 1-10. Get Target Version Sequence

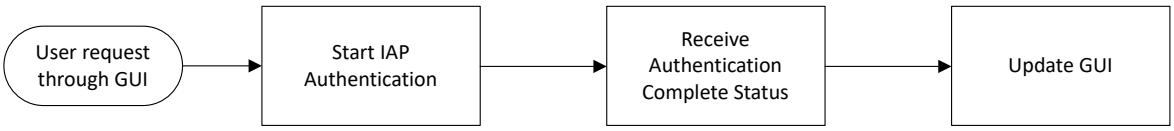


When the `Get Target Version` request is received from the GUI, the Data Store module sends an instruction to the Data Gen module via Mi-V to generate and send a control packet with `Get Target Version` command. After receiving the status and the version number, the version number is passed to the GUI.

1.1.3.1.3 Start IAP Authentication

The following figure shows the sequence of `Start IAP Authentication` command execution.

Figure 1-11. Start IAP Authentication Sequence

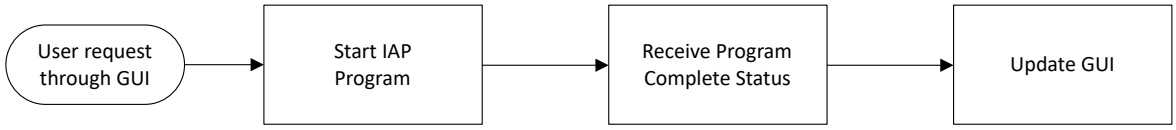


When the `Authenticate IAP Image` request is received from the GUI, the Data Store module sends an instruction to the Data Gen module via Mi-V to generate and send a control packet with `Start IAP Authentication` command. After receiving the status and the version number, they are passed to the GUI.

1.1.3.1.4 Start IAP Program

The following figure shows the sequence of `Start IAP Program` command execution.

Figure 1-12. Start IAP Program Sequence



When the `Program the Image` request is received from the GUI, the Data Store module sends an instruction to the Data Gen module via Mi-V to generate and send a control packet with `Start IAP Program` command. After receiving the status and the version number, the updated version number is passed to the GUI.

1.1.3.2 Data Gen

Based on the requests from the Data Store module, the Data Gen module generates Ethernet packets with the appropriate control and status data to be transmitted to IAP Target.

The Ethernet packet consists of the following fields:

- Destination Address (DA): Custom destination address used for the demo design.
- Source Address (SA): MAC address associated with the port through which the Ethernet packet is transmitted.
- Ethernet Type: Reserved Ethernet type used for the demo design.
- Command: Each activity has a specific command value for a better readability.
- Length: Payload length value of control and data packets.
- Offset: TBD
- Payload: Control packet can have a maximum payload length of 64 bytes and the data packet can have a maximum payload length of 1024 byes.
- FCS: Four-byte frame check sequence as defined in clause 4 of [IEEE 802.3].
- EOF: End of packet.

The Data Gen module uses this custom packet structure and fields while generating control and data Ethernet packets in this demo design. The Control Ethernet packet of length 92 bytes carries the necessary information to IAP Target for the following activities.

- Data Transfer request: This activity happens frequently to check the status of IAP Target for starting the data transfer. IAP Initiator keeps sending this request packet until the response packet is received.
- Last Burst and Last Burst size: This packet indicates the end of the SPI image to the IAP Target.

The data packet of length 1052 bytes carries the SPI image to the target, which is stored in the External SPI Flash of IAP Target. Control packet is also used to send image authentication and program requests. For any given data transfer, a total of 2 KB data (2 data packets) are transmitted every time.

The following table lists the byte format of a control and data packet.

Table 1-1. Ethernet Packet Structure (Control and Data)

Byte Number	Size	Field	Value
1-8	8 bytes	Destination Address	48'hFFFF_FFFF_FFFF
		Source Address	16'hD0AB

.....continued			
Byte Number	**Size**	**Field**	**Value**
9-16	8 bytes	Source Address	32'hD5576610
		Ethernet Type	16’h0700
		Command	16’h0004 – Control 16’h0005 – Payload 16’h0007 – IAP Authenticate 16’h0008 – IAP Program 16’h0009 – Send version Number 16’h000A – Clear all interrupts
16-23	8 bytes	Zero’s	32’h00000000
		Length	0x0040 – 64 bytes
			0x0400 – 1024 bytes
		Offset	16’h0000
24-87	36 bytes	Payload (Control Information)	64’h0000000000000000 – For Control Info packet {Burst size, Last burst info} – For Authenticate, Program packets
Data Packet			
24-1047	1024 bytes	Payload (IAP Data)	1024 Bytes of IAP Data
Last 4	4 bytes	FCS	Core10GMAC computes the FCS and appends to the Frame before transmitting out.

1.1.3.3 Data Parser

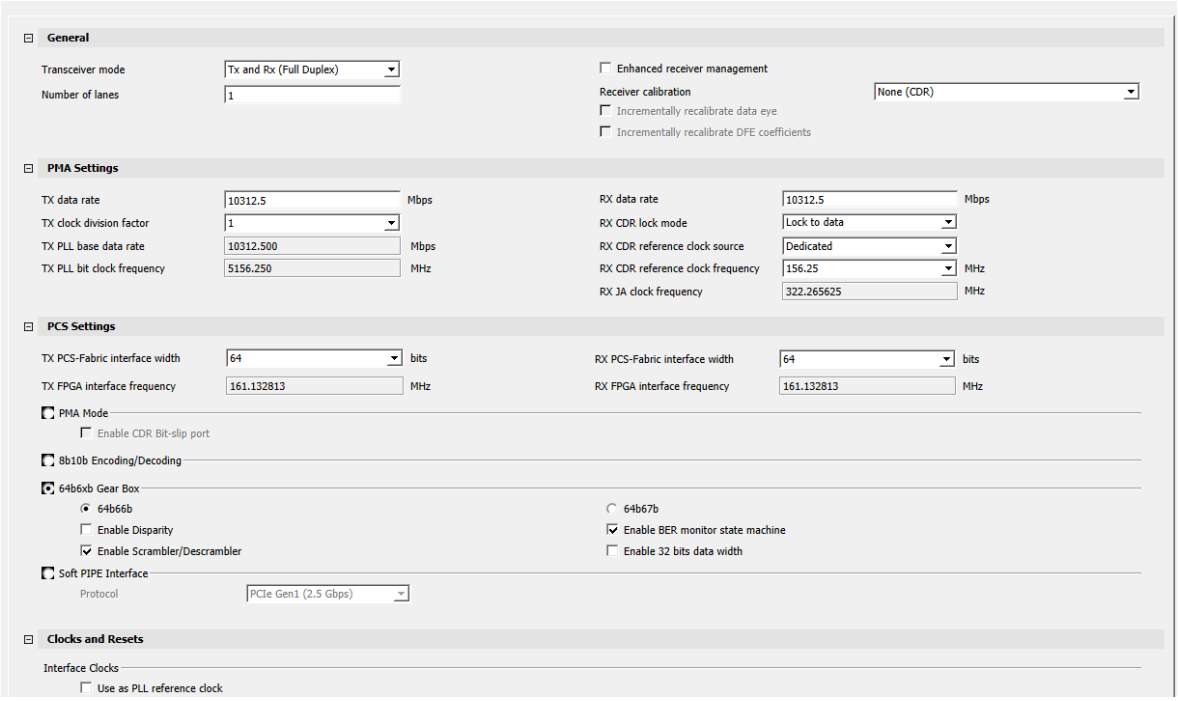
This module parses and decodes the data coming from IAP Target and updates the status to the user. Based on the command field, one of the following status information is decoded:

- Start data transfer acknowledgment from IAP target.
- Image write complete status.
- Authentication complete status and Authentication error code.
- Version number received status and version number information.

1.1.4 PF_XCVR_0 (IAP Initiator)

The PolarFire high-speed transceiver (PF_XCVR_ERM) is a hard IP block and supports data rates from 500 Mbps to 12.7 Gbps. In this demo, PF_XCVR_ERM is configured for the data rate of 10312.5 Mbps. It is configured with a CDR reference clock of 156.25 MHz with lock to data selected as the CDR lock mode. The PCS of the transceiver is interfaced with CORE10GMAC. It is configured for the 64/66b mode with scrambler/descrambler enabled. The scrambler, which is self-synchronizing, generates sufficient transitions to aid data and clock recovery at the CDR. The following figure shows the transceiver interface configuration.

Figure 1-13. PF_XCVR Configuration

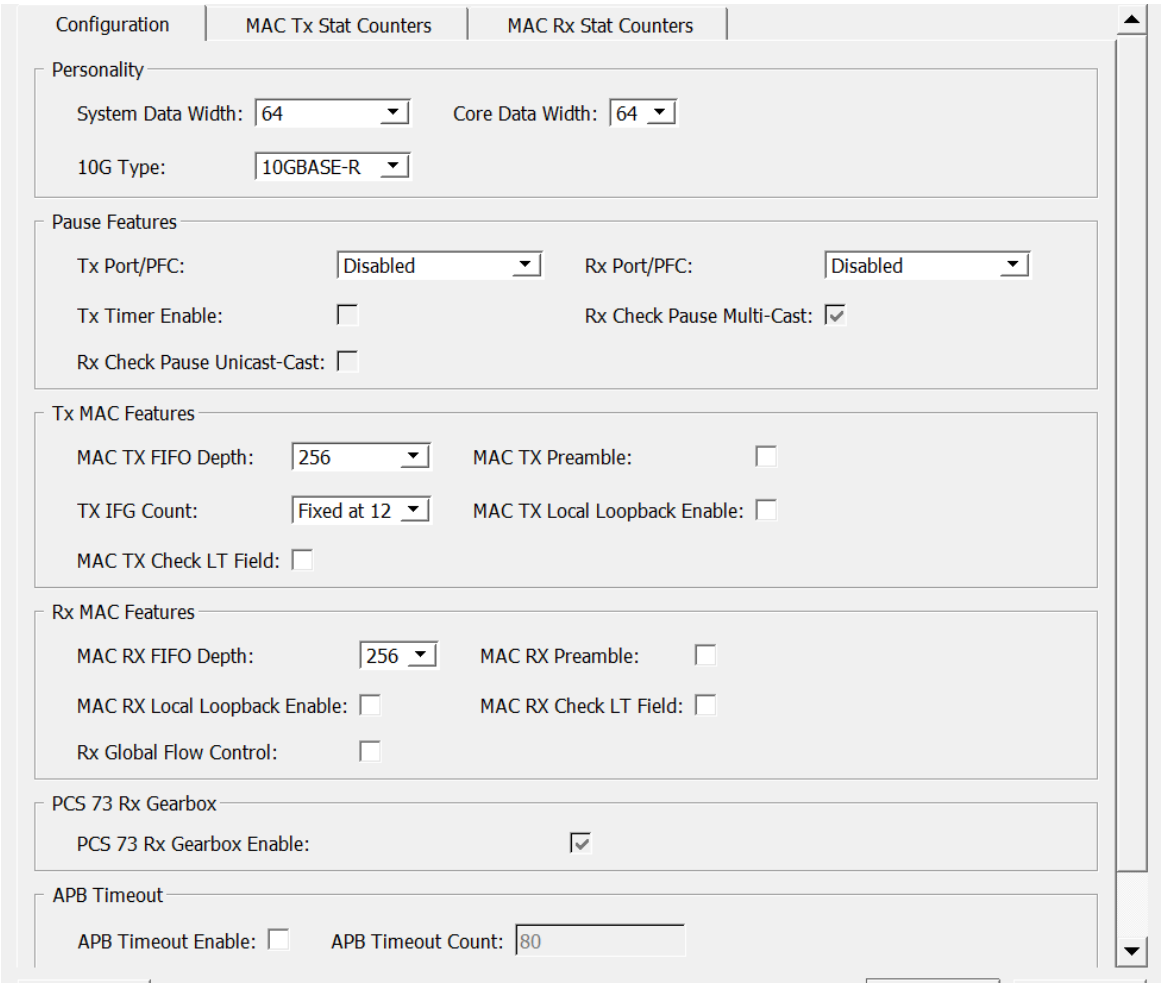


1.1.5 CORE10GMAC0 (IAP Initiator)

The Core10GMAC IP is configured for 10GBASE-R mode with a core data width of 64 bits. Core data width is the width of the data path connected to the transceiver interface. The system data width, that is, the width of the interface to the user logic, is configured as 64 bits. The Tx and Rx Pause features are disabled, and both the MAC TX FIFO depth and MAC RX FIFO depth are set to 256.

The CORE10GMAC IP is configured as shown in the following figure.

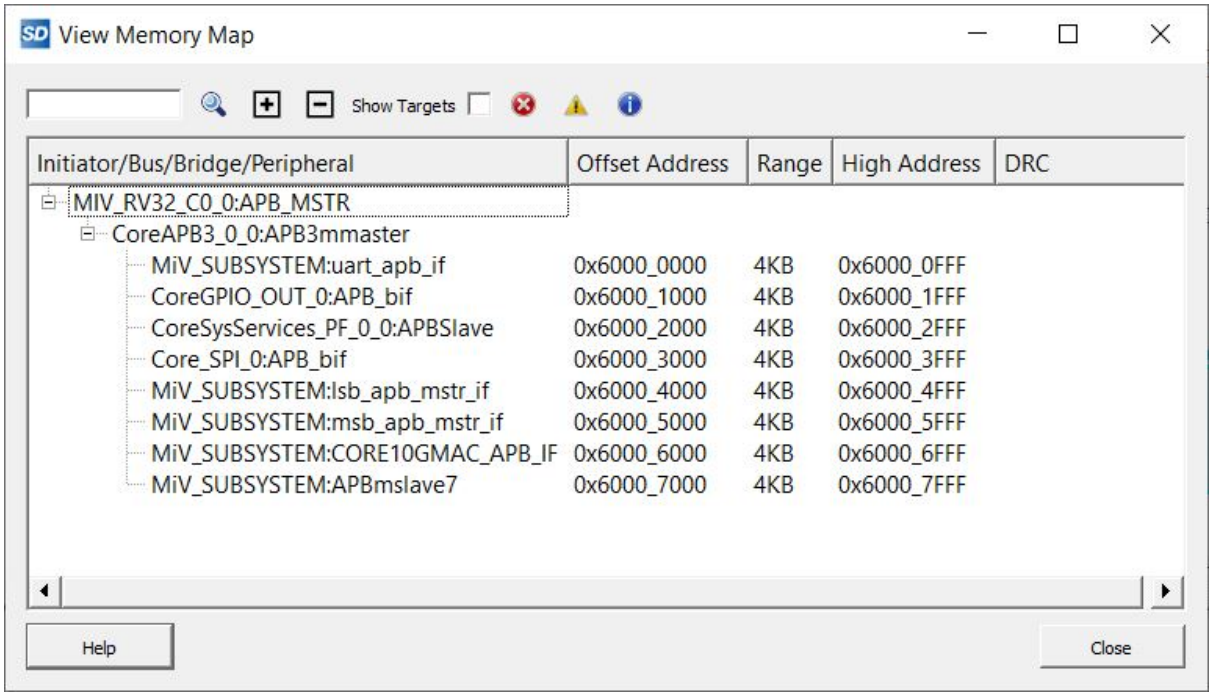
Figure 1-14. CORE10GMAC Configuration



1.1.6 Memory Map (IAP Initiator)

The Mi-V soft processor accesses the peripherals per the following memory map.

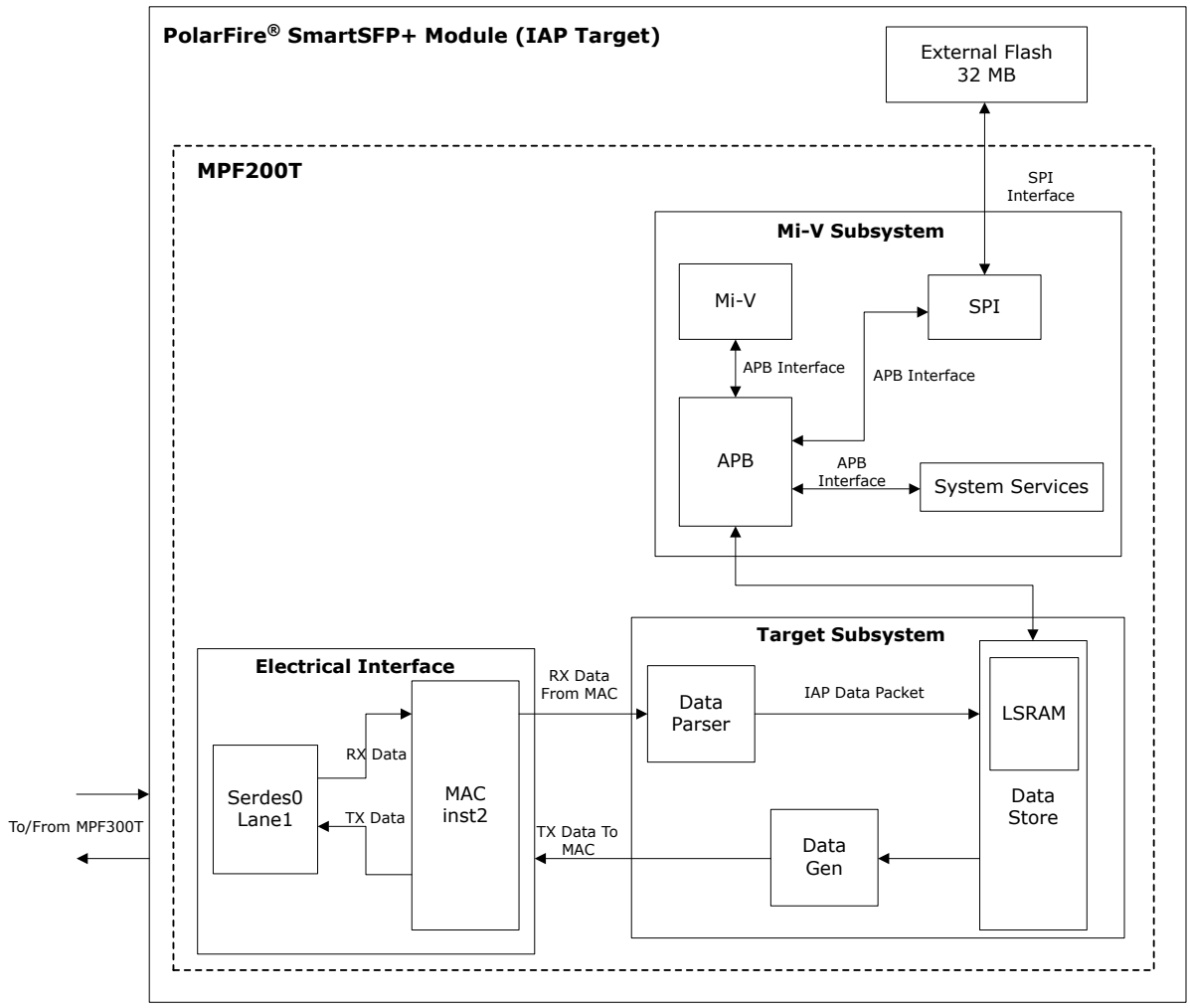
Figure 1-15. Memory Map (IAP Initiator)



1.2 IAP Target

The following figure shows the functional blocks of IAP Target.

Figure 1-16. IAP Target Block Diagram



The main components of IAP Target are as follows:

- 1.2.1. Target Subsystem
- 1.2.2. Mi-V Subsystem (Target Version)
- 1.2.3. Electrical Interface

1.2.1 Target Subsystem

The Target subsystem is implemented to handle the following functions:

- Receive and store the incoming IAP image data from IAP Initiator in the 32 MB external SPI Flash.
- Authenticate the stored image.
- Execute IAP.

The Target subsystem consists of the following sub-blocks:

- 1.2.1.1. Data Parser (IAP Target)
- 1.2.1.2. Data Store (IAP Target)
- 1.2.1.3. Data Gen (IAP Target)

1.2.1.1 Data Parser (IAP Target)

This module parses the incoming data coming from IAP Initiator based on the Ethernet packet type (0x0700), and passes the same to the Data Store module. The Data Parser module operates in the 156 MHz clock domain.

1.2.1.2 Data Store (IAP Target)

This module decodes the parsed data from the data parser and generates the appropriate interrupts and maintains the register space for the Mi-V soft processor to execute instructions and read the status.

IAP Target's Data Store module stores 2 KB of incoming IAP data from IAP Initiator in LSRAM and generates an interrupt for the Mi-V soft processor to indicate that the data is ready. After receiving the interrupt, the Mi-V soft processor reads the IAP data from LSRAM and writes the IAP data to the external SPI Flash.

The following interrupts are generated:

- Authenticate the image data.
- Program the authenticated image.
- Send the target version number.
- Clear the commands.

1.2.1.3 Data Gen (IAP Target)

This module generates Ethernet packets with the appropriate control packets to be transmitted to IAP Initiator based on the instruction from the Mi-V soft processor. The packet structure is described in Table 1-2.

In this demo design, the Data Gen module uses the custom packet structure and fields to generate Control Ethernet packets. The Control Ethernet packet of length 92 bytes carries the necessary information to IAP Target for activities listed in the following table.

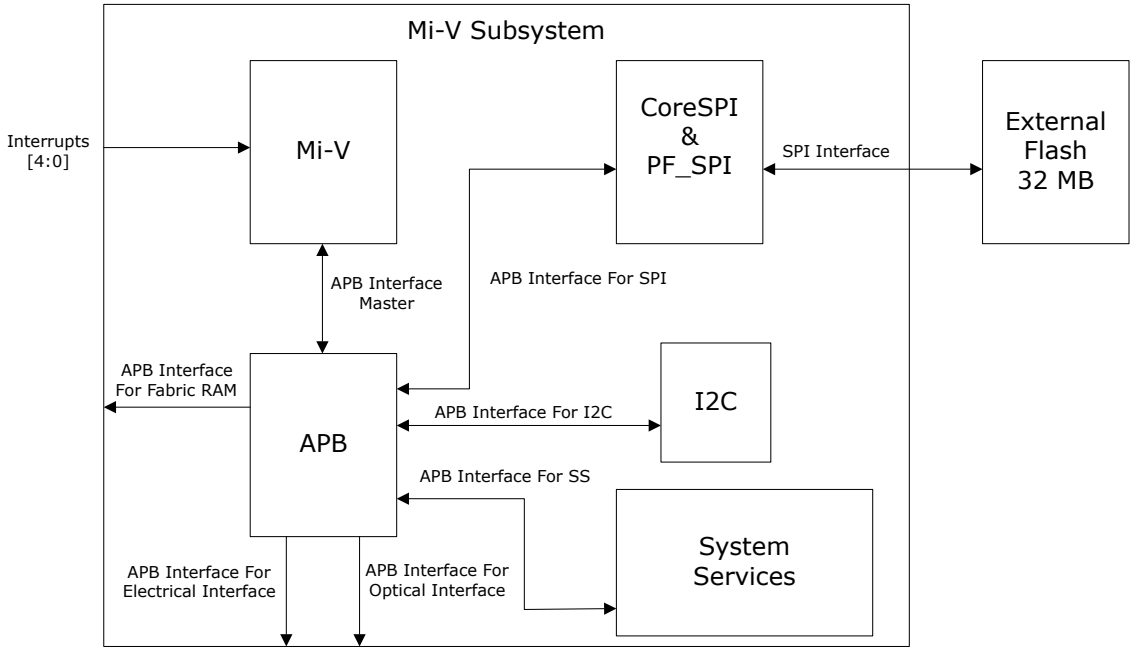
Table 1-2. Ethernet Packet Structure (Control)

Byte Number	Size	Field	Value
1-8	8 bytes	• Destination Address • Source Address	• 48'hFFFF_FFFF_FFFF • 16'hD0AB
9-16	8 bytes	Source Address	32'hD5576610
		Ethernet Type	16'h0700
		Command	• 16'h0001 – Image write complete • 16'h0002 – Authenticate complete • 16'h0003 – target version number sent • 16'h0004 – Control information • 16'h0006 – IAP Authentication Complete • 16'h000B – IAP Mode ON 16'h000C – Loopback Mode ON
16-23	8 bytes	Zero's	32'h00000000
		Length	0x0040 – 64 bytes
		Offset	16'h0000
24-87	64 bytes	Payload (Control Information)	64'h0000000000000000 (for all packets)
Last 4	4 bytes	FCS (Frame Check Sequence)	Core10GMAC computes the FCS and appends to the Frame before transmitting out.

1.2.2 Mi-V Subsystem (Target Version)

The following figure the main components of the Mi-V subsystem.

Figure 1-17. Mi-V Subsystem (IAP Target)



The Mi-V subsystem executes the following functions:

1. Send version number: Interrupt to send the version number to the initiator.
2. Data ready: Interrupt to read the IAP data from LSRAM.
3. Start IAP authentication: Interrupt to perform authentication.
4. Start IAP programming: Interrupt to perform IAP.
5. Clear the sequence: Interrupt to clear the commands and status.

The Mi-V subsystem includes the Mi-V soft processor, CoreGPIO, and CoreSPI IPs, which are configured in the same way as in IAP Initiator.

Note: The CoreAPB3 in IAP Target is configured to have an extra slot for I^2C.

1.2.3 Electrical Interface

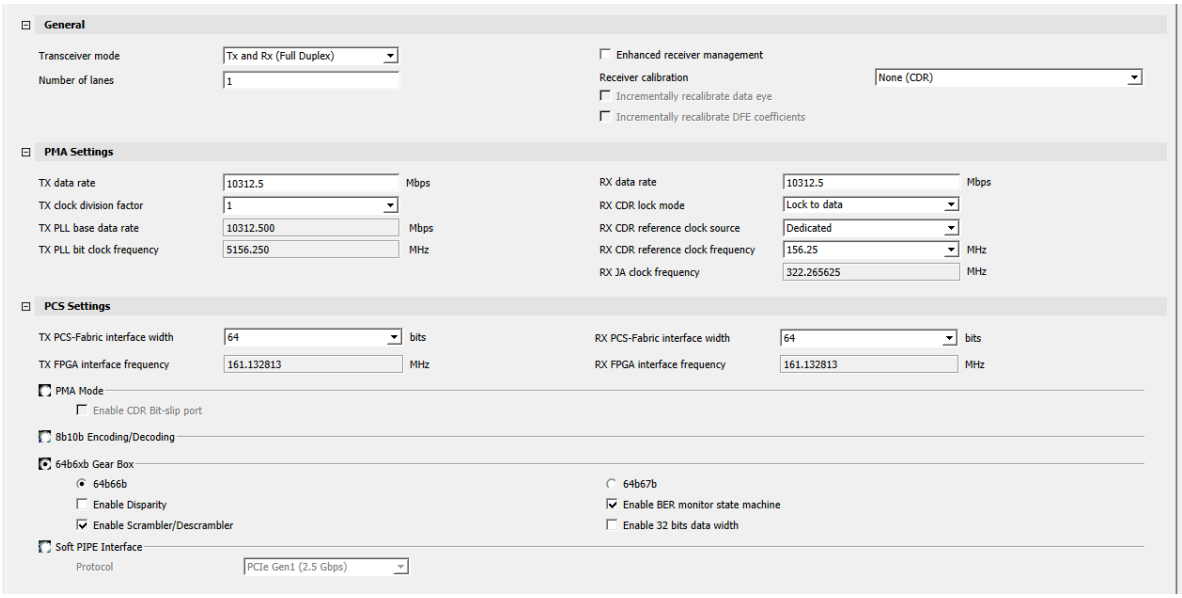
The electrical interface consists of the followings components:

- 1.2.3.1. PF_XCVR_ERM_C2 (IAP Target)
- 1.2.3.2. CORE10GMAC_C0 (IAP Target)

1.2.3.1 PF_XCVR_ERM_C2 (IAP Target)

The PF_XCVR_ERM IP is configured as shown in the following figure.

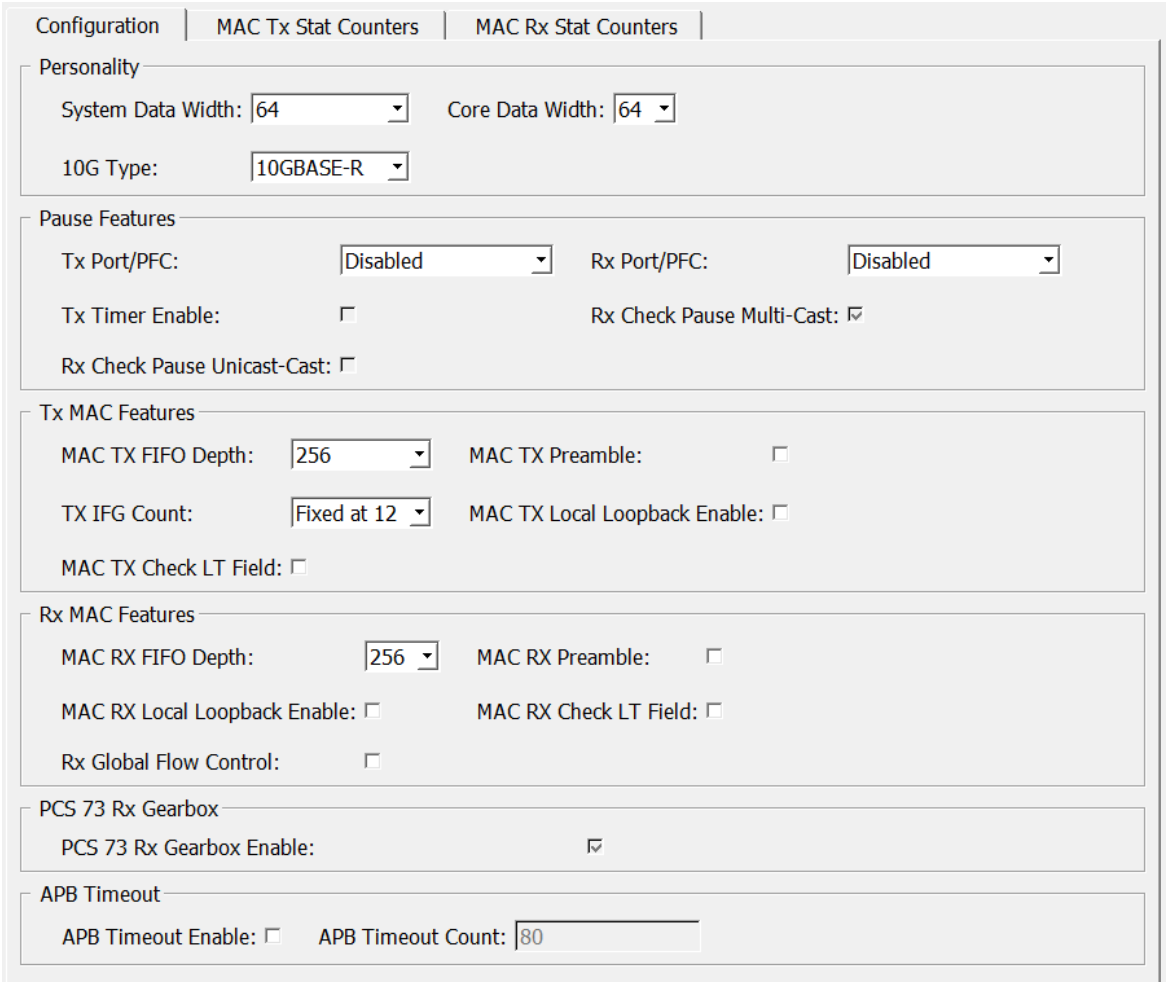
Figure 1-18. PF_XCVR_ERM_C2 Configuration



1.2.3.2 CORE10GMAC_C0 (IAP Target)

The CORE10GMAC IP is configured as shown in the following figure.

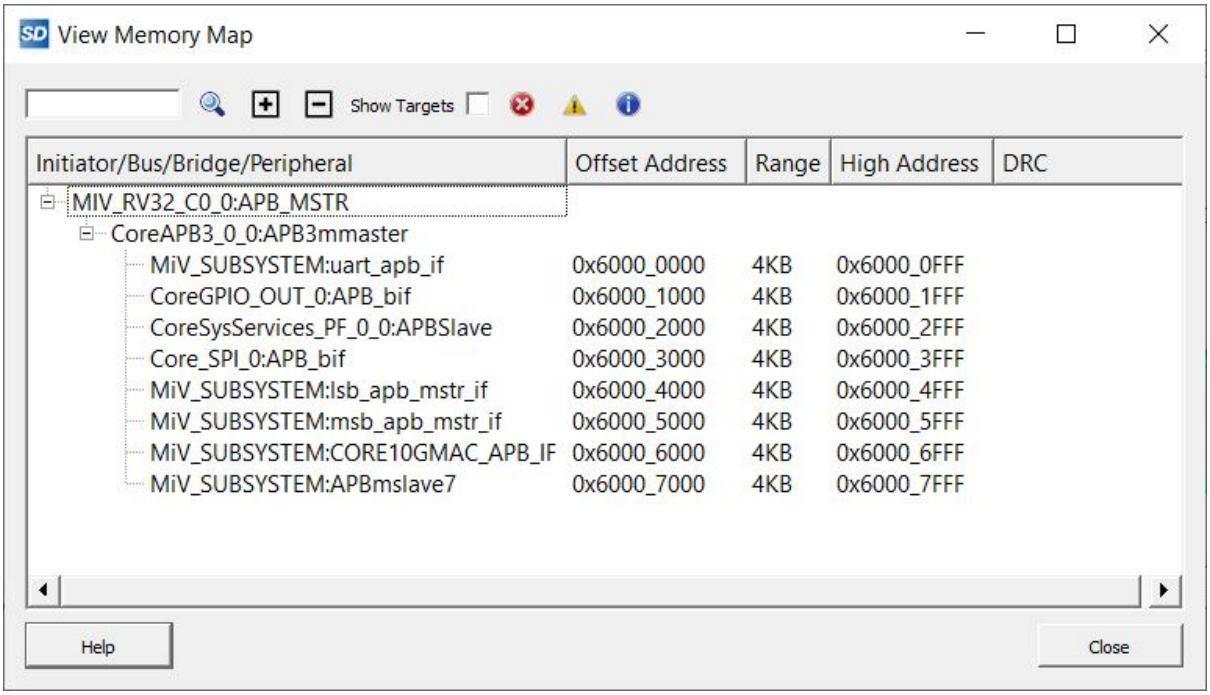
Figure 1-19. CORE10GMAC_C0 Configuration



1.2.4 Memory Map (IAP Target)

The Mi-V soft processor accesses the peripherals per the following memory map.

Figure 1-20. Memory Map (IAP Target)



1.3 Clocking Structure

The following table lists all the clocks used in the IAP Initiator design.

Table 1-3. Clocking Structure (IAP Initiator)

Clock Name	Frequency in MHz	Instance Name
Clk_156	156.25	CORE10GMAC0_0
		IAP_INITIATOR_SUBSYSTEM_0
Clk_50	50	CORE10GMAC0_0
		MiV_SUBSYSTEM_0
		UART_IF_0
TX_CLK_R	161.133	CORE10GMAC0_0
RX_CLK_R	161.133	CORE10GMAC0_0

The following table lists all the clocks used in the IAP Target design.

Table 1-4. Clocking Structure (IAP Target)

Clock name	Frequency (MHz)	Instance Name
O_MAC_SYS_CLK	156.25	CORE10GMAC0_0
		IAP_TARGET_SUBSYSTEM_0
		FIFO_SUBSYSTEM_0

.....continued		
Clock name	**Frequency (MHz)**	**Instance Name**
O_Clk_50	50	CORE10GMAC0_0
		MiV_SUBSYSTEM_0
		IAP_TARGET_SUBSYSTEM_0
O_I2C_clk	1	MiV_SUBSYSTEM_0
TX_CLK_R	161.133	CORE10GMAC0_0
RX_CLK_R	161.133	CORE10GMAC0_0

1.4 Reset Structure

IAP Initiator and IAP Target use the following Reset scheme:

- reset_156 : For all the signals generated and synchronized in the 156 MHz domain.
- reset_50: For all the signals generated and synchronized in the 50 MHz domain.

2. SPI Programming Images

The following `.spi` programming images are generated and exported for programming external SPI Flash of IAP Initiator.

Table 2-1. SPI Programming Images

Image Name	Version	Silicon Signature	Image Index	Start Address in SPI Flash
GOLDEN	0	N/A	0	0x00000400
IAP_IMAGE	2	0x12345678	2	0x00A00000

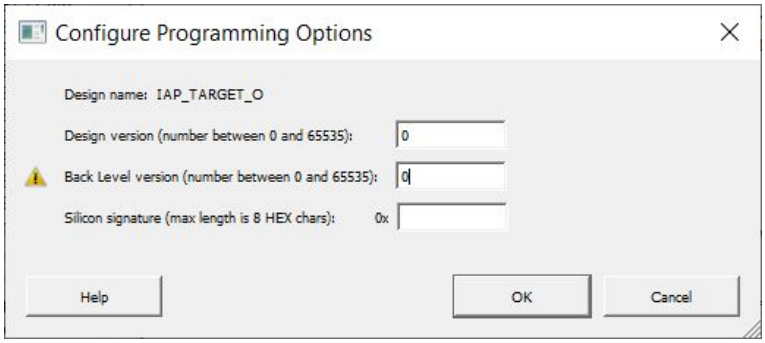
Important:

1. The GOLDEN image cannot contain any security or silicon signature information. The GOLDEN image is generated from `<$download_directory>\mpf_an4568_v2022p1_df\Libero_Projects\IAP_Target\iap_target_o.prjx`.
2. The IAP image is generated from `<$download_directory>\mpf_an4568_v2022p1_df\Libero_Projects\IAP_Target\iap_target_c\iap_target_c.prjx`.

2.1 Programming Options for GOLDEN Image

The GOLDEN image is programmed on the MPF200T device when the IAP process is aborted for any error. The GOLDEN image is generated from `iap_target_o.prjx` project. The following figure shows the programming options configured for the GOLDEN `.spi` image. To open this window, select **Configure Programming Options** from **Libero Design Flow** > **Program Design**.

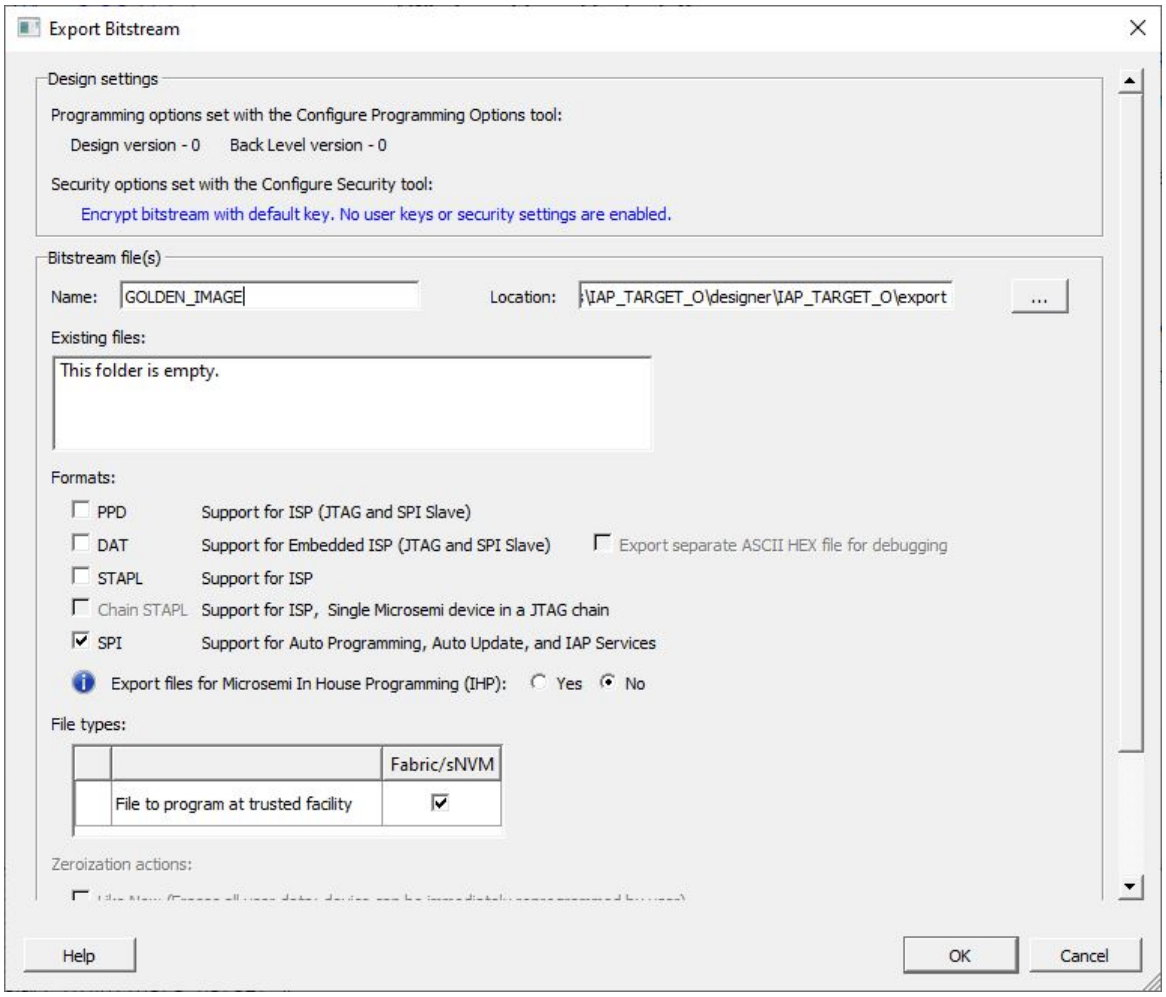
Figure 2-1. Programming Options of the Golden Image



2.2 Exporting the GOLDEN Image

The following figure shows the **Export Bitstream** window that is used to export the GOLDEN image. To open this window, select **Configure Programming Options** from **Libero Design Flow** > **Handoff Design for Production**.

Figure 2-2. Export Bitstream Window (GOLDEN Programming Image)

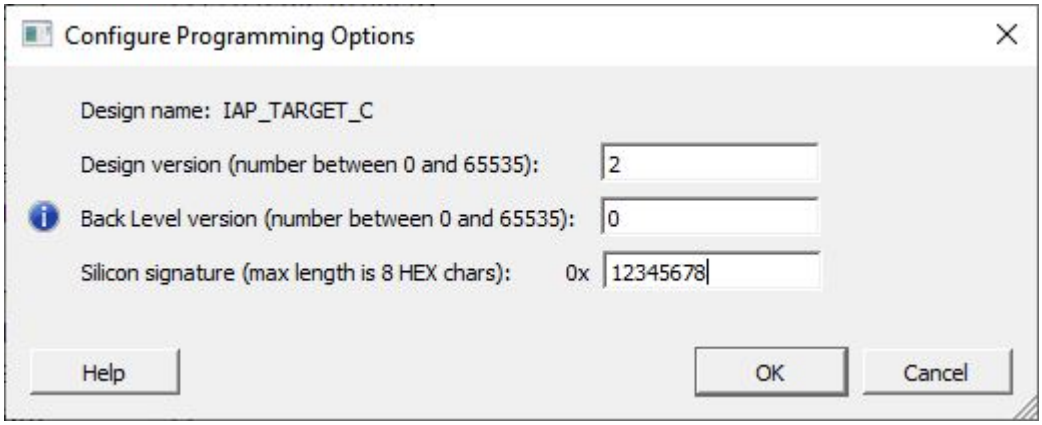


Important: The bitstream must be exported after it is generated.

2.3 Programming Options for IAP Image

The IAP image is the image that is programmed on the MPF200T device for IAP update. The IAP image is generated from `iap_target_c.prjx` project. The following figure shows the programming options configured for the IAP `.spi` image. To open this window, select **Configure Programming Options** from **Libero Design Flow** > **Program Design**.

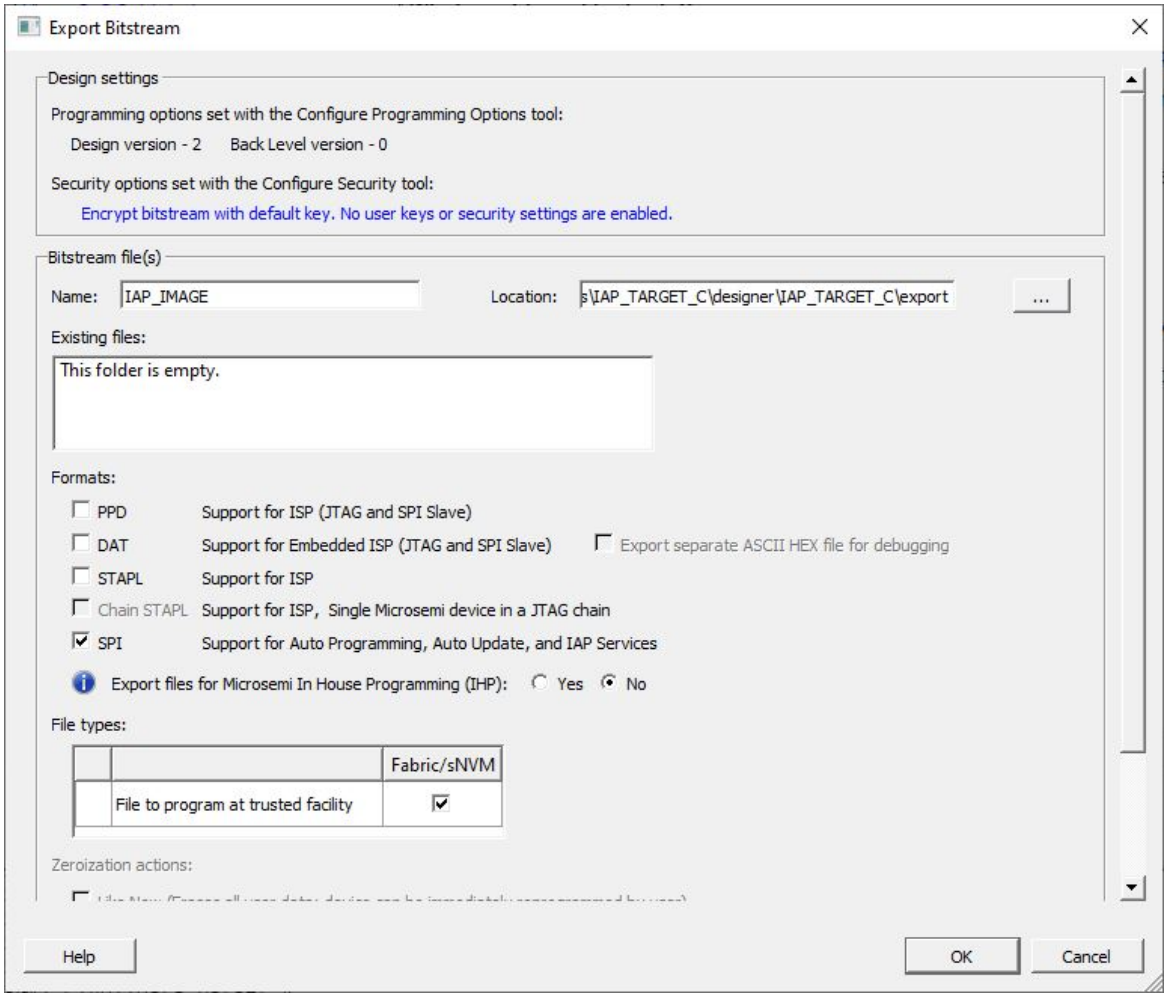
Figure 2-3. Programming Options of the IAP Image



2.4 Exporting the IAP Image

The following figure shows **Export Bitstream** window that is used to export the IAP image. To open this window, select **Configure Programming Options** from **Libero Design Flow** > **Handoff Design for Production**.

Figure 2-4. Export Bitstream Window (IAP Image)



Important: The bitstream must be exported after it is generated.

3. Demo Requirements

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

Table 3-1. Demo Requirements

Requirement	Description
Hardware and Accessories	
• PolarFire SmartSFP+ Module (MPF200T device) - Pre-release version • PolarFire Evaluation Kit (MPF300T device) - Rev D	
USB A to mini-B cable (two cables)	Required for the following: • FPGA programming. • UART interface for the terminal prompt from Linux.
External FlashPro6 Programmer	Required for programming the PolarFire MPF200T device on the SmartSFP+ Module.
Power adapters (two adapters)	12V, 5A
Host PC	A host PC with USB port.
Utility Software	
SMART SFP+ IAP GUI	Application used for initiating IAP request to PolarFire Evaluation Kit which writes the IAP bitstream to the SPI Flash on the PolarFire SmartSFP+ module.

4. Demo Prerequisites

Before you start:

- Download the design files for programming from the following link: www.microchip.com/en-us/application-notes/AN4568
- Download and install Libero® SoC Design Suite from Libero SoC Software Downloads.
- See AN4364: PolarFire FPGA SFP+ Module Application Note for hardware details.

5. Setting Up the Demo

Setting up the demo involves the following steps:

1. 5.1. Setting Up the Hardware
2. 5.2. Programming the MPF300T Device
3. 5.3. Programming the MPF200T Device
4. 5.4. Generating the IAP Update Design
5. 5.5. Installing the GUI

5.1 Setting Up the Hardware

This section describes the steps to setup the hardware for programming the PolarFire devices and running the demo.

Follow these steps to setup the hardware:

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

Table 5-1. Jumper Setting on PolarFire Evaluation Board

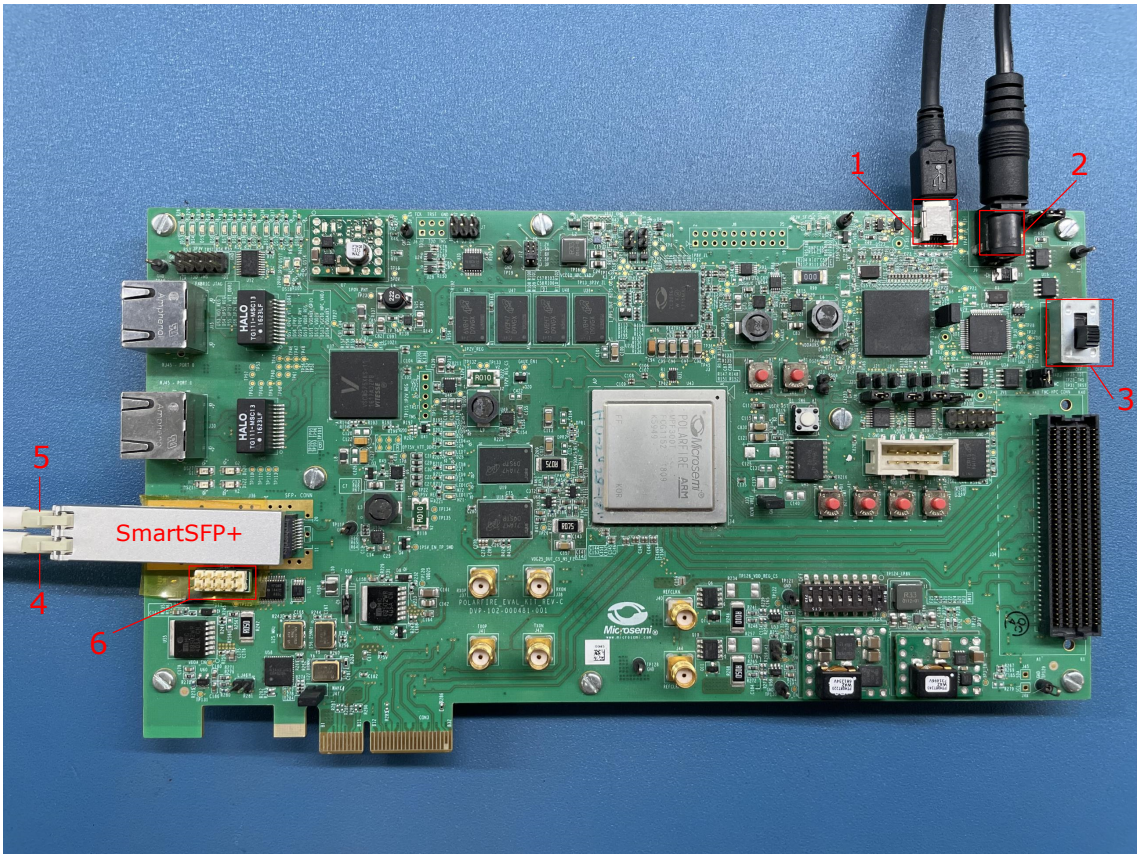
Jumper	Setting
J18, J19, J20, J21, and J22	Close pins 2 and 3 for programming through FTDI.
J28	Close pins 1 and 2 for programming through the on-board FlashPro5.
J4	Close pins 1 and 2 for switching the power manually using SW3.
J12	Close pins 3 and 4 for 2.5V.

2. Connect the power supply cable to the J9 connector on the board.
3. Connect the USB cable from the host PC to J5 (FTDI port) on the board.
4. Power-up the board using the SW3 slide switch.
5. Insert the PolarFire SmartSFP+ module in the SFP+ slot on the PolarFire Evaluation board.
   The SmartSFP+ module is switched ON and draws the required power supply via the SFP+ interface.
6. Connect the external FlashPro6 programmer from the JTAG header on the SmartSFP+ module to the host PC.
   This is required for programming the MPF200T device on the SmartSFP+ module.

This concludes the hardware setup.

The following figure shows the hardware setup.

Figure 5-1. Board Setup



1. USB Connector 2. Power Connector 3. ON/OFF Switch 4. SFP+ RX 5. SFP+ TX 6. JTAG Header

The following sections provide information on how to program the `iap_target_o` design on the MPF200T device, and to program `iap_initiator_top` design on the MPF300T device.

5.2 Programming the MPF300T Device

This section describes the steps to re-create and program the IAP Initiator design on the MPF300T device by executing the TCL script available in the design files. Before you start, ensure that the version of IP cores in the Libero SoC vault is exactly the same as specified in `<$download_directory>\mpf_an4568_v2022p1_df\INITIATOR\common.tcl`.

Follow these steps:

1. Ensure that the hardware is setup as described in 5.1. Setting Up the Hardware.
2. On the host PC, launch Libero SoC.
3. Select **Project** > **Execute Script**.
4. Select **Browse** and select `script.tcl` from `<$download_directory>\mpf_an4568_v2022p1_df\INITIATOR\script.tcl`.
5. Select **Run**. After successful execution of the TCL script, the following steps are completed:
   a. The user application is selected and the TCM initialization client is generated with the user application.
   b. The SPI Flash client is generated with the IAP image.
   c. The IAP Initiator design is programmed on the MPF300T device.
   d. The on-board SPI Flash is programmed with Golden and IAP image.

e. The Libero SoC project for IAP Initiator is created (`iap_initiator.prjx`).

Tip: For more information about TCL commands, see *Libero® SoC TCL Command Reference Guide*. Contact Microchip Support for any queries about running the TCL script.

This concludes the programming of the MPF300T device.

5.3 Programming the MPF200T Device

This section describes the steps to re-create and program the IAP Target design on the MPF200T device by executing the TCL script available in the design files. Before you start, ensure that the version of IP cores in the Libero SoC vault is exactly the same as specified in `<$download_directory>\mpf_an4568_v2022p1_df\TARGET_O\common.tcl`.

Follow these steps:

1. Ensure that the hardware is setup as described in 5.1. Setting Up the Hardware.
2. On the host PC, launch Libero SoC.
3. Select **Project** > **Execute Script**.
4. Select **Browse** and select `script.tcl` from `<$download_directory>\mpf_an4568_v2022p1_df\TARGET_O\script.tcl`.
5. Select **Run**. After successful execution of the TCL script, the following steps are completed:
   a. The user application is selected and the TCM initialization client is generated with the user application.
   b. The SPI Flash client is generated with the IAP Golden image.
   c. The IAP Target design is programmed on the MPF200T device.
   d. The on-board SPI Flash is programmed with Golden and IAP image.
   e. The Libero SoC project for IAP Target is created (`iap_target_o.prjx`).

Tip: For more information about TCL commands, see *Libero® SoC TCL Command Reference Guide*. Contact Microchip Support for any queries about running the TCL script.

This concludes the programming of the MPF200T device.

5.4 Generating the IAP Update Design

This section describes the steps to create IAP update design for the IAP Target by executing the TCL script available in the design files. In this design, the SPI Flash client is generated with the IAP image. This image is used for IAP. Before you start, ensure that the version of IP cores in the Libero SoC vault is exactly the same as specified in `<$download_directory>\mpf_an4568_v2022p1_df\TARGET_C\common.tcl`.

Follow these steps:

1. Ensure that the hardware is setup as described in 5.1. Setting Up the Hardware.
2. On the host PC, launch Libero SoC.
3. Select **Project** > **Execute Script**.
4. Select **Browse** and select `script.tcl` from `<$download_directory>\mpf_an4568_v2022p1_df\TARGET_C\script.tcl`.
5. Select **Run**. After successful execution of the TCL script, the Libero SoC project for IAP update is created (`iap_target_c.prjx`).

Tip: For more information about TCL commands, see *Libero® SoC TCL Command Reference Guide*. Contact Microchip Support for any queries about running the TCL script.

5.5 Installing the GUI

Follow these to install the GUI:

1. Extract the `mpf_an4568_v2022p1_df.zip` file.
2. Double click the `setup.exe` file from `<$download_directory>\mpf_an4568_v2022p1_df\GUI`.
3. Follow the instructions on the installation wizard. After successful installation, the SmartSFP demo appears on the start menu of the host PC.

This concludes installing the demo. See 6. Running the Demo for running the demo.

6. Running the Demo

This demo is divided into the following parts:

1. 6.1. Executing IAP
2. 6.2. Executing Data Loopback

6.1 Executing IAP

This section describes the steps to use GUI for running the IAP demo. The GUI is used to perform the following:

- Fetch the IAP Target design version.
- Transfer SPI IAP image.
- Authenticate the transferred IAP image.
- Program the IAP sequence on MPF200T device.

Follow these steps to run the demo:

1. Ensure that the hardware is setup as described in 5.1. Setting Up the Hardware.
2. Power ON the PolarFire Evaluation board by sliding the SW3 switch.
3. On the host PC, launch the GUI. Before connecting to the PolarFire Evaluation board, the GUI shows the default values of the image size as 6183952 Bytes and target version number fields as `0`. The GUI title also remains in grey colour, as shown in the following figure.

   Figure 6-1. GUI Before Connection

The GUI automatically detects the COM port and establishes a serial connection. After the PolarFire Evaluation board is connected, the GUI confirms the connection changing the GUI title to blue color. Also, the **Target Version** is updated with the actual target version number. The default image size is 6183952 Bytes. After connecting, the GUI switches to the IAP mode as shown in the following figure.

Figure 6-2. GUI After Connection

4. Select the **Image Transfer** option and click **Initiate** to start the transfer of the IAP image to the SPI Flash of IAP Target. The image transfer status is displayed as shown in the following figures.

Figure 6-3. Image Transfer Progress

Figure 6-4. Image Transfer Complete

5. Select **Authentication IAP Image** and click **Initiate** to start the authentication of the IAP image by IAP Target. When **Authentication IAP Image** is selected, IAP Initiator sends appropriate commands to initiate the authentication process at IAP Target's end. The authentication status is displayed as shown in the following figures.

Figure 6-5. Authenticating IAP Image

Figure 6-6. Authentication Successful

Note: Authentication can be selected only after the completion of Image Transfer showing a green tick.

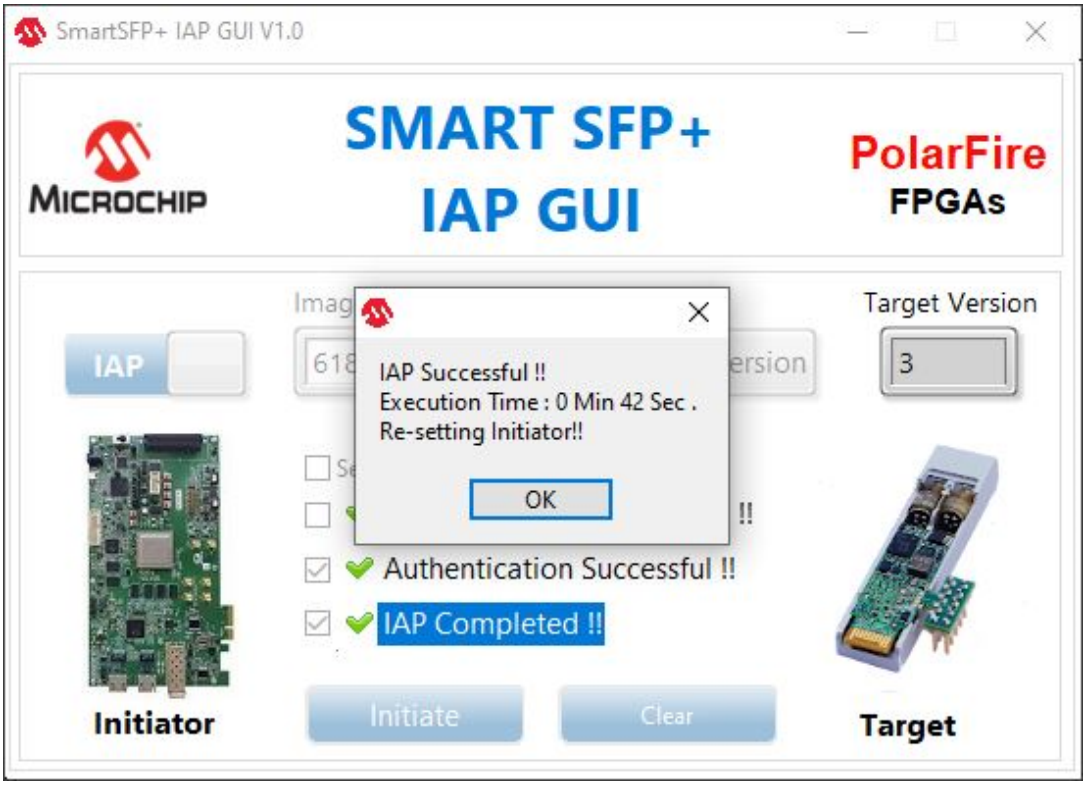
6. Select the **Program the Image** option and click **Initiate**. IAP Initiator sends the appropriate command to IAP Target device to initiate the IAP sequence. The programming progress is displayed as shown in the following figure.

Figure 6-7. Programming Status

Note: The **Program the Image** command can only be selected along with **Authenticate IAP Image** command.

The execution time and updated target version are displayed as status, see the following figure.

Figure 6-8. Programming Successful



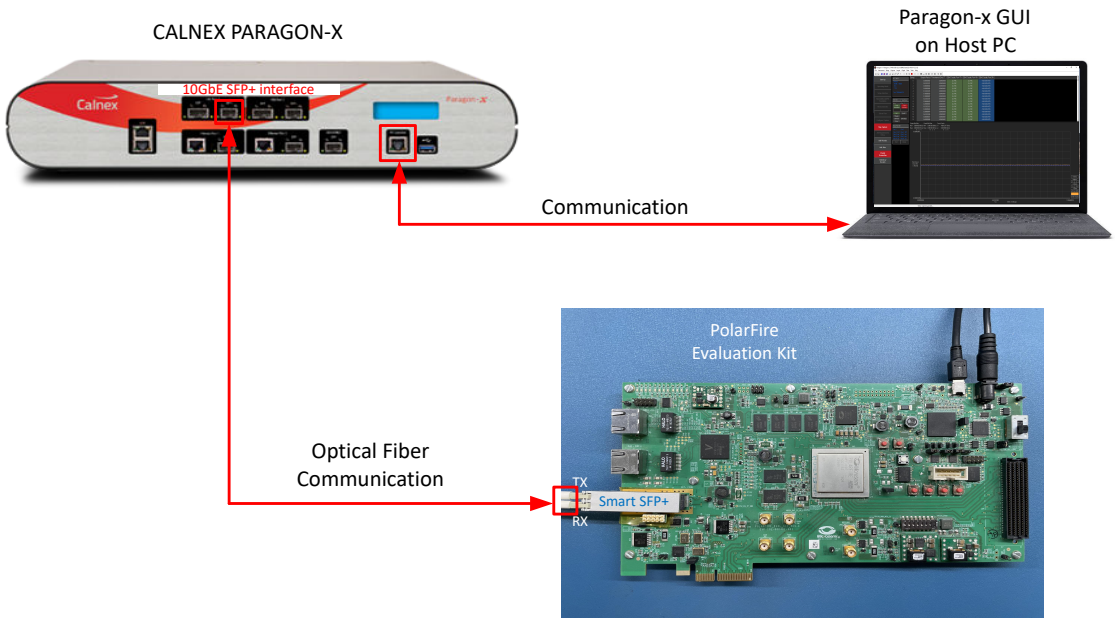
7. To run all the three IAP commands in sequence, enable the **Select All** check box and click initiate.

This concludes executing IAP. See the next section to execute data loopback.

6.2 Executing Data Loopback

Before executing data loopback, a 10G Ethernet traffic generator must be connected to the SmartSFP+ as shown in the following figure.

Figure 6-9. Loopback Setup

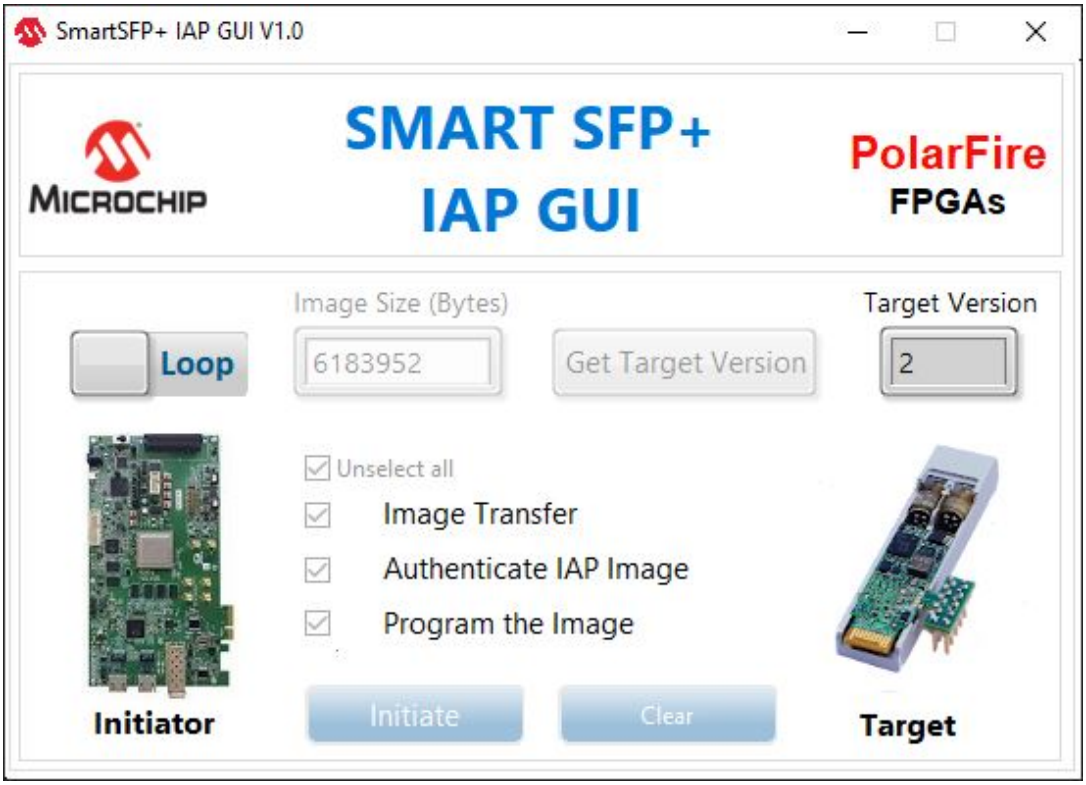


Note: In this example, Paragon-X Tester was used for generating 10G Ethernet packets and validating the design. Use an equivalent 10G Ethernet packet generator.

Follow these steps to use the GUI for executing the data loopback:

1. Click the mode switch button to enable the Data Loopback mode. When the Loop mode is selected all the controls related to the IAP mode are disabled and greyed out. See the following figure.

Figure 6-10. Switch From IAP to Loop Mode



Once the Data Loopback mode is enabled, the 10G Ethernet packets are looped back from the SmartSFP+ module to the Traffic generator.

7. Revision History

Revision	Date	Description
A	05/2022	The first publication of the document.

Microchip FPGA Support

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

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

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

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

Microchip Information

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

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, WinPath, 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, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscient Code Generation, PICDEM,

ISBN: 978-1-6683-0400-6

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