PolarFire FPGA System Services

AN4663

Introduction (Ask a Question)

System services are System Controller actions initiated from the FPGA design using the PF_SYSTEM_SERVICES core. The System Controller hard block in PolarFire® FPGAs provide various system services. The PF_SYSTEM_SERVICES core issues service requests to the System Controller and fetches the relevant data.

Table of Contents

Introduction....1

1. PolarFire FPGA System Services....3
   - 1.1. PF_SYSTEM_SERVICES Core Overview....3
   - 1.2. Design Requirements....4
   - 1.3. Prerequisites....5
   - 1.4. Demo Design....5
   - 1.5. Clocking Structure....11
2. Libero Design Flow....13
   - 2.1. Synthesize....14
   - 2.2. Place and Route....15
   - 2.3. Verify Timing....15
   - 2.4. Generate FPGA Array Data....15
   - 2.5. Configure Design Initialization Data and Memories....15
   - 2.6. Configure Programming Options....18
   - 2.7. Generate Bitstream....19
   - 2.8. Run PROGRAM Action....19
   - 2.9. Export FlashPro Express Job....20
3. Setting up the Serial Terminal Program - PuTTY....21
4. Running the Demo....23
5. Appendix 1: Programming the Device Using FlashPro Express....29
6. Appendix 2: Device Certificate Information....31
7. Appendix 3: Query Security....33
8. Appendix 4: Debug Information....34
9. Appendix 5: Digest Information....35
10. Appendix 6: Running the TCL Script....36
11. Appendix 7: References....37
12. Revision History....38

Microchip FPGA Support....39

Microchip Information....39
- The Microchip Website....39
- Product Change Notification Service....39
- Customer Support....39
- Microchip Devices Code Protection Feature....39
- Legal Notice....40
- Trademarks....40
- Quality Management System....41
- Worldwide Sales and Service....42

1. PolarFire FPGA System Services (Ask a Question)

This document describes how to run the system services using the demo design. The following table lists the system services used in the demo.

Table 1-1. System Services in the Demo

Service Category	Services
Device and Data Services	Read Device Serial Number Read Device User-code Read Device Design-info
Design and Data Security Services	Read Device Certificate Read Digest Query security Read Debug Information Digital signature Secure NVM services PUF Emulation Nonce service

The demo design includes the Mi-V soft processor, which initiates the system service requests and enables the PF_SYSTEM_SERVICES core to access the System Controller. For more information about the system services design implementation, and the necessary blocks and IP cores instantiated in Libero® SoC, see 1.4. Demo Design.

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

- **Using the pre - generated .job file**: To program the device using the `.job` file provided along with the demo design, see 5. Appendix 1: Programming the Device Using FlashPro Express.
- **Using Libero SoC**: To program the device using Libero SoC, see 2. Libero Design Flow.

The demo design can be used as a reference to build a fabric design with the system services feature.

1.1 PF_SYSTEM_SERVICES Core Overview (Ask a Question)

System controller actions are initiated by the fabric logic through the System Service Interface (SSI) of the System Controller. The fabric logic requires the PF_SYSTEM_SERVICES core for initiating the system services. A service request interrupt to the system controller is triggered when the fabric user logic writes a 16-bit system service descriptor to the SSI. The lower seven bits of the descriptor specify the service to be performed. The upper nine bits specify the address offset (0–511) in the 2 KB mailbox RAM. The mailbox address specifies the service-specific data structure used for any additional inputs or outputs for the service. The fabric logic must write additional parameters to the mailbox before requesting a system service.

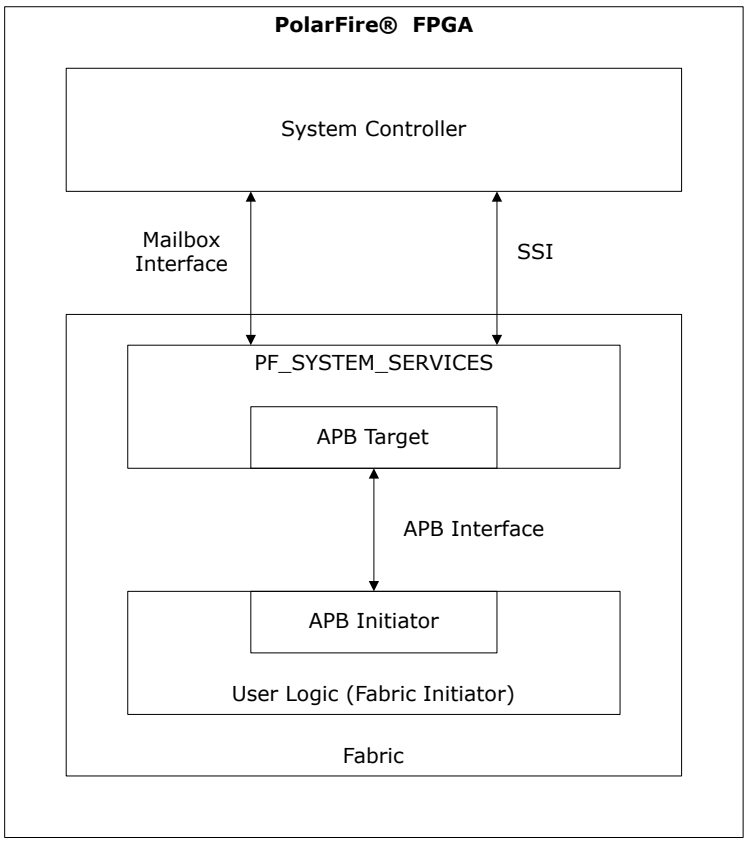
The following table lists the system service descriptor bits.

Table 1-2. System Services Descriptor

Descriptor Bit	Value
15:7	MBOXADDR
6:0	SERVICEID

SSI consists of an asynchronous command-response interface that transfers a system service command from the fabric master to the system controller and the status from the system controller to the fabric master. The following figure shows how the PF_SYSTEM_SERVICES interfaces with the fabric logic.

Figure 1-1. Core System Services IP Interfacing with Fabric User Logic



The system services driver and the sample SoftConsole project are available in `driver-examples/CoreSysServices-PF` folder in the Bare Metal Examples on GitHub. For more information on sNVM secure access, see the following:

- The "Secure Non-Volatile Memory" section in PolarFire FPGA and PolarFire SoC FPGA Security User Guide.
- The `CoreSysServices-PF` driver example in the Bare Metal Examples on GitHub.

In this demo, the sample SoftConsole project is migrated to SoftConsole and the application file `main.c` is modified to provide the user options.

1.2 Design Requirements (Ask a Question)

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

Table 1-3. Design Requirements

Requirement	Version
Operating System	Windows® 7, 8.1, or 10
Hardware	
PolarFire® Evaluation Kit (MPF300TS_1FCG1152I)	Rev D or later
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 Design Suite	
SoftConsole	

.....continued	
Requirement	**Version**
Serial Terminal Emulation Program	PuTTY or HyperTerminal www.putty.org

Important:

- Any serial terminal emulation program can be used. PuTTY is used in this demo.
- Libero SmartDesign and configuration screen shots shown in this guide are for illustration purpose only. Open the Libero design to see the latest updates.

1.3 Prerequisites (Ask a Question)

Before you begin:

1. Download and install Libero SoC (as indicated in the website for this design) on the host PC from the following location.Libero SoC Documentation
2. For demo design files download link: For Evaluation Kit www.microchip.com/en-us/application-notes/AN4663

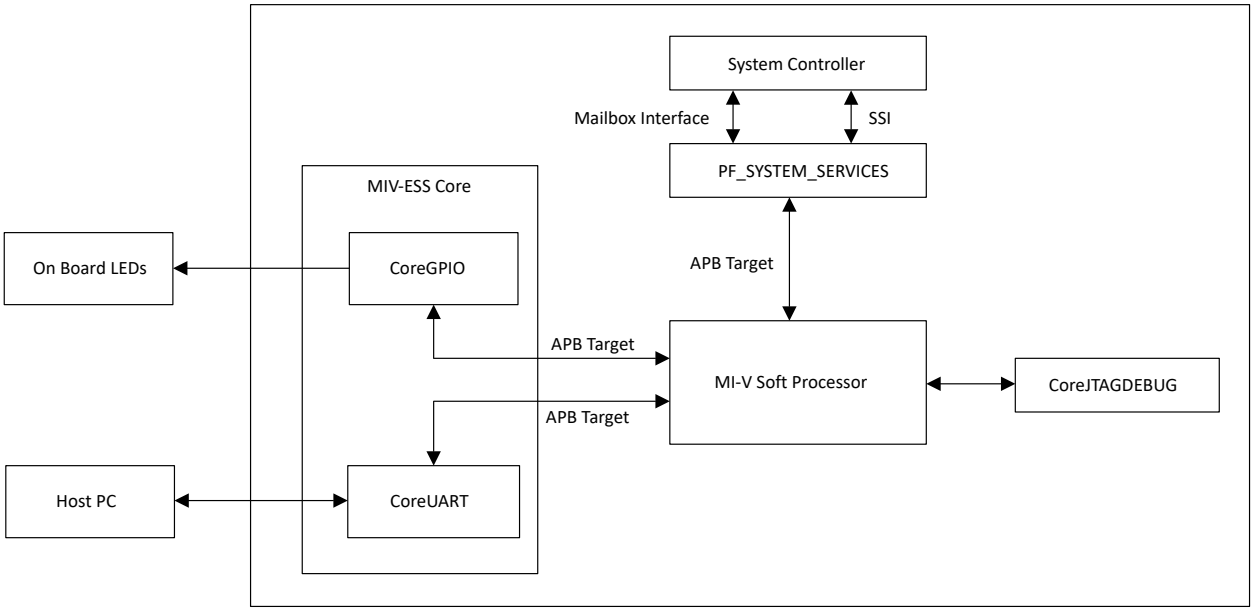
1.4 Demo Design (Ask a Question)

The following steps describe the data flow in the demo design:

1. The host PC sends the system service requests to MIV_ESS_C0 CoreUARTapb block through the UART Interface.
2. The Mi-V soft processor initializes the system controller using the PF_SYSTEM_SERVICES and sends the requested system service command to the system controller.
3. The system controller executes the system service command and sends the relevant response to the PF_SYSTEM_SERVICES over the mailbox interface.
4. The Mi-V processor receives the service response and forwards the data to the UART interface.

The following figure shows the block diagram of the system services design.

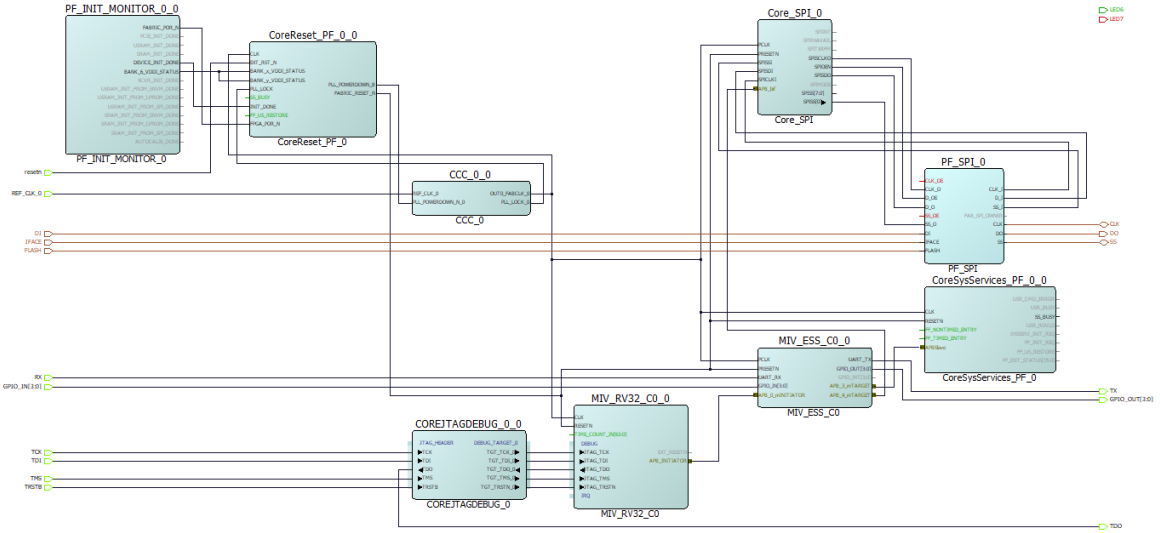
Figure 1-2. System Services Design Block Diagram



1.4.1 Design Implementation (Ask a Question)

The following figure shows the top-level Libero design of the PolarFire system services design.

Figure 1-3. Top-Level Libero Design



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

Table 1-4. I/O Signals

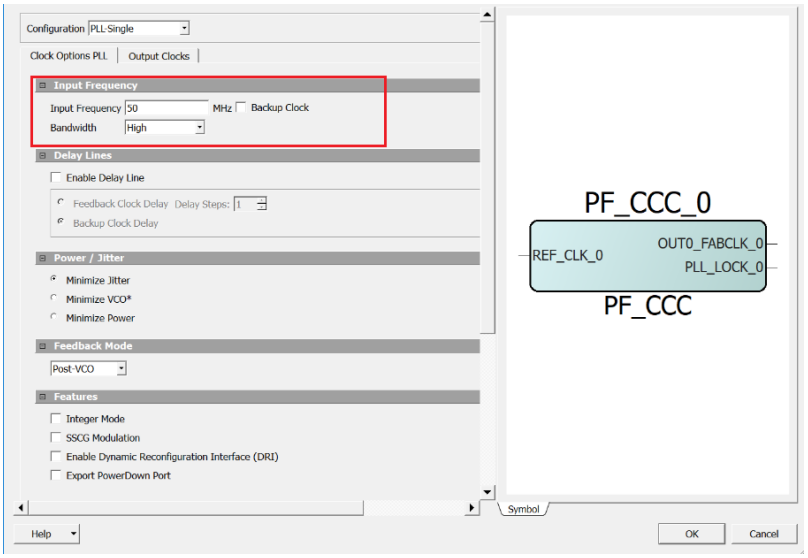
Signal	Description
REF_CLK_0	Input 50 MHz clock from the onboard 50 MHz oscillator
resetn	Onboard reset push button for the PolarFire® device
RX	Input signals received from the serial UART terminal
TX	Output signals transmitted to the serial UART terminal
GPIO_OUT[3:0]	Onboard LED outputs

1.4.1.1 PF_CCC_0 Configuration (Ask a Question)

The PolarFire Clock Conditioning Circuitry (CCC) block takes an input clock of 50 MHz from the onboard oscillator and generates a 83.33 MHz fabric clock to the Mi-V processor subsystem and other peripherals.

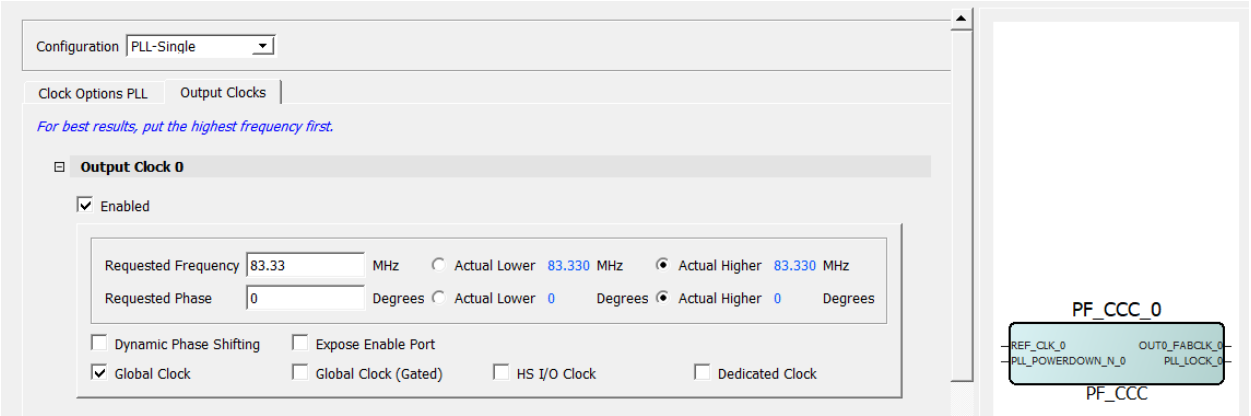
The following figures show the input and output clock configurations.

Figure 1-4. PF_CCC_0 Input Clock Configuration



The following figure shows the PF_CCC_0 output clock configuration. This design uses a 83.33 MHz system clock for configuring the APB peripherals.

Figure 1-5. PF_CCC_0 Output Clock Configuration



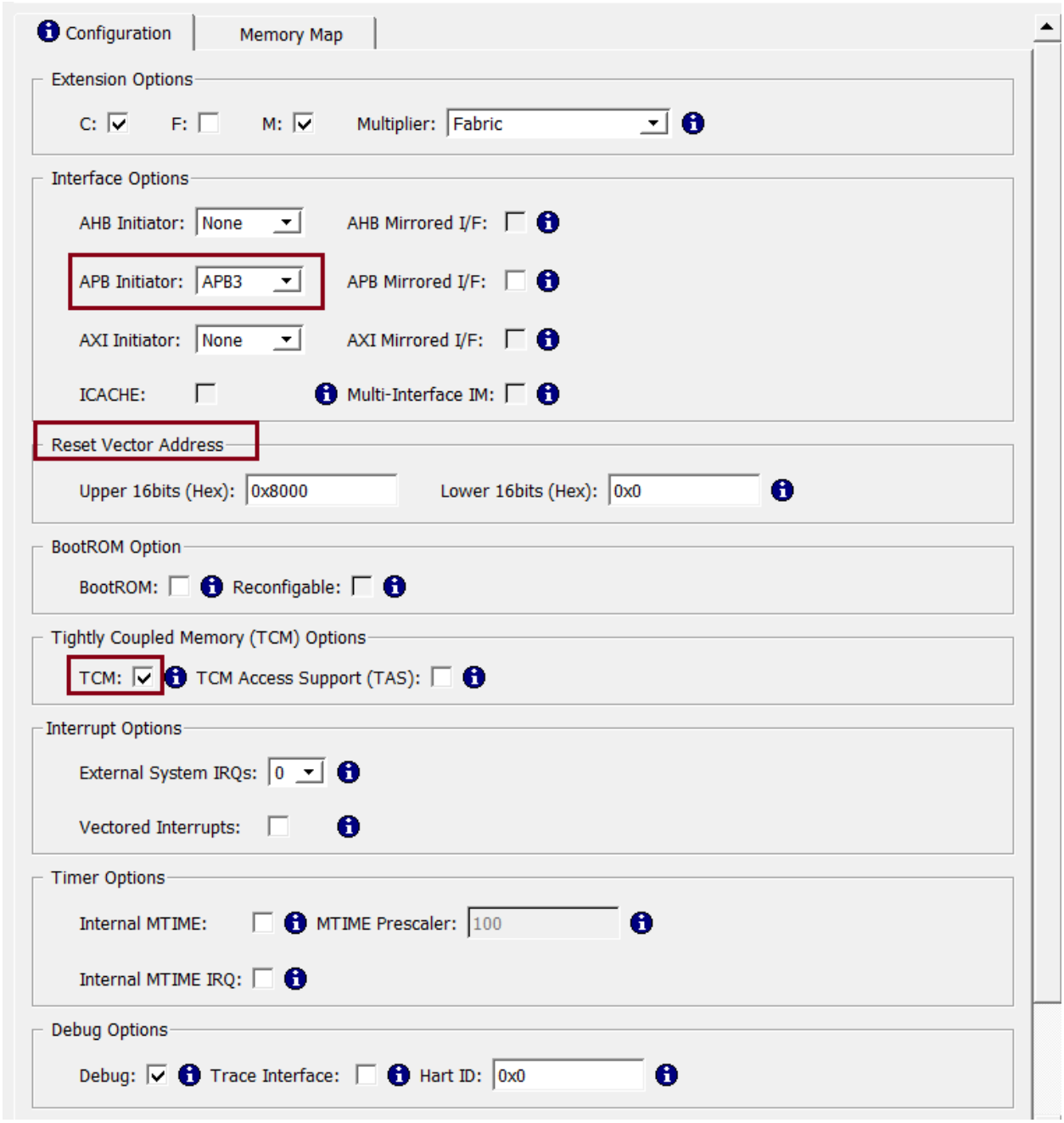
1.4.1.2 Mi-V Soft Processor Configuration (Ask a Question)

The Mi-V soft processor default Reset Vector Address is 0x8000_0000. After the device reset, the processor executes the application from TCM, which is mapped to 0x80000000, hence the Reset Vector Address is set to 0x80000000, see Figure 1-6.

TCM is the main memory of the Mi-V processor. It gets initialized with the user application from µPROM.

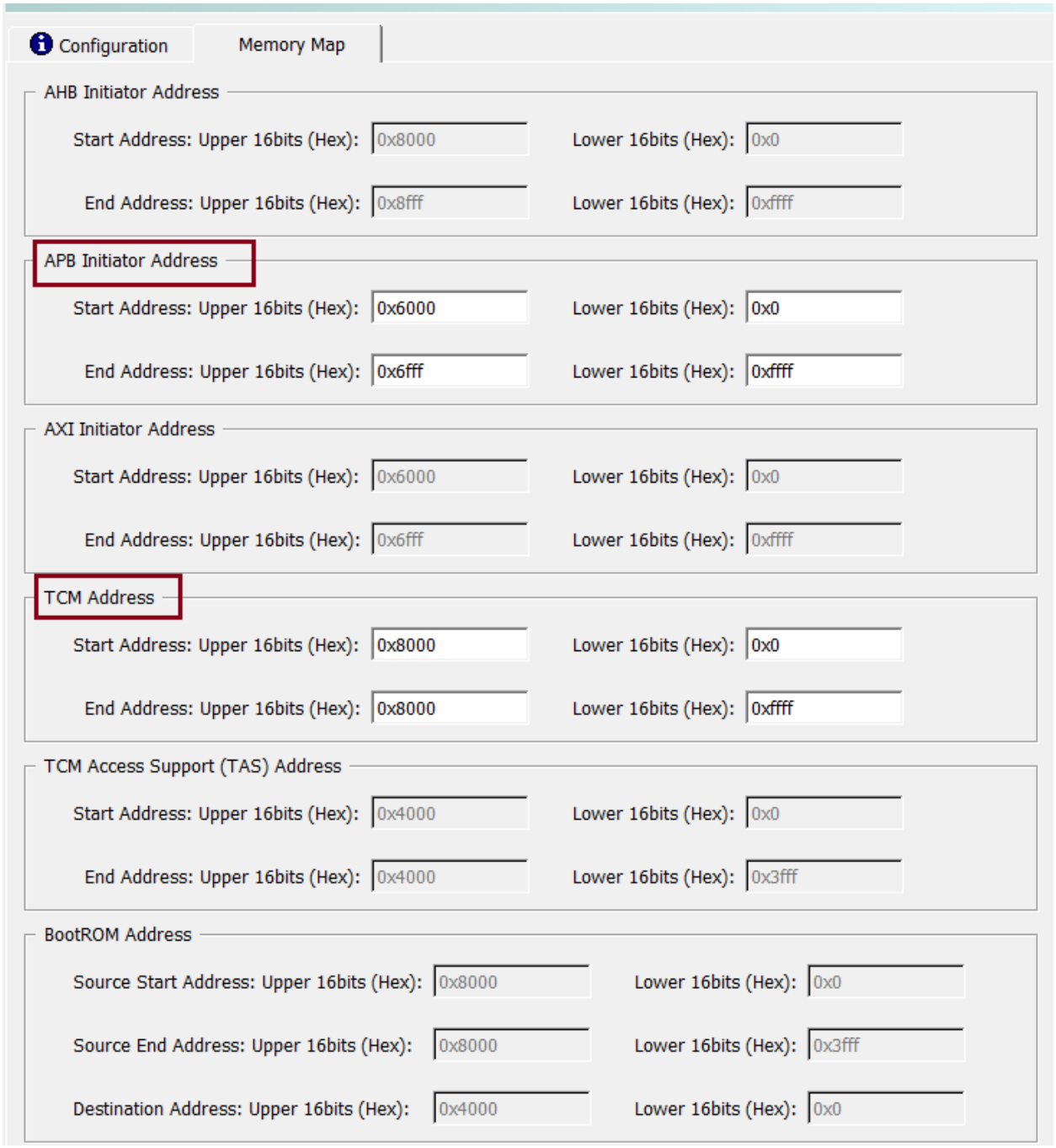
In the Mi-V processor memory map, the 0x8000_0000 to 0x8000_FFFF range is defined for TCM memory interface and the 0x6000_0000 to 0x6FFF_FFFF range is defined for APB3 I/O interface.

Figure 1-6. Mi-V RV32 Configuration



Memory depth: This field is set to 16384 words to accommodate an application of up to 64 KB into the TCM. The present application is less than 50 KB, so that it can fit into either sNVM or µPROM. In this demo, µPROM is selected as a data storage client, see the following figure.

Figure 1-7. Mi-V RV32 Configuration1



1.4.1.3 MIV ESS Core (Ask a Question)

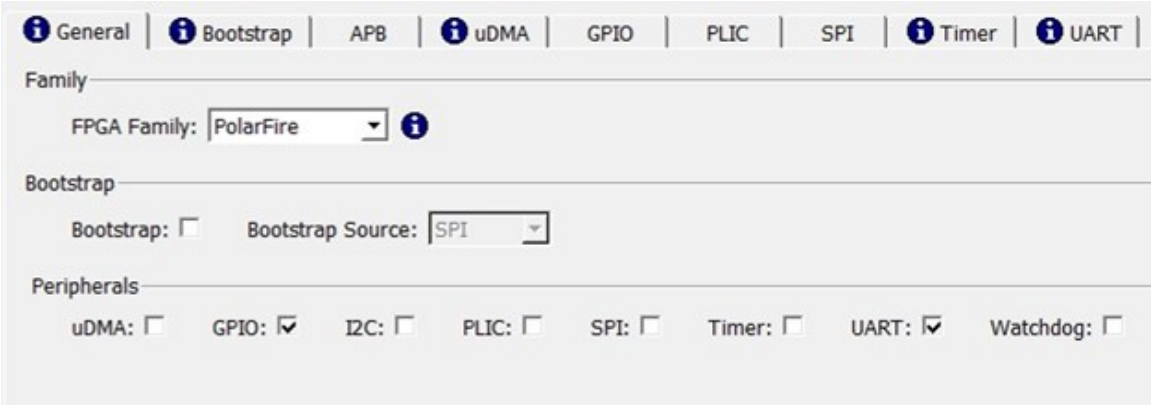
The MIV_ESS is a MI-V ecosystem IP core available for the Microchip FPGA and System- on-Chip (SoC) FPGA device families. The core is a multi-featured, highly-configurable, Extended Subsystem (ESS), which supports bothbootstrap and base peripherals. It is specifically designed to use with the MIV_RV32 soft processor. APB interface connects the PF_SYSTEM_SERVICES, CoreSPI as an external peripheral.

Instantiating MiV ESS Core

To instantiate the MiV ESS core, perform the following steps:

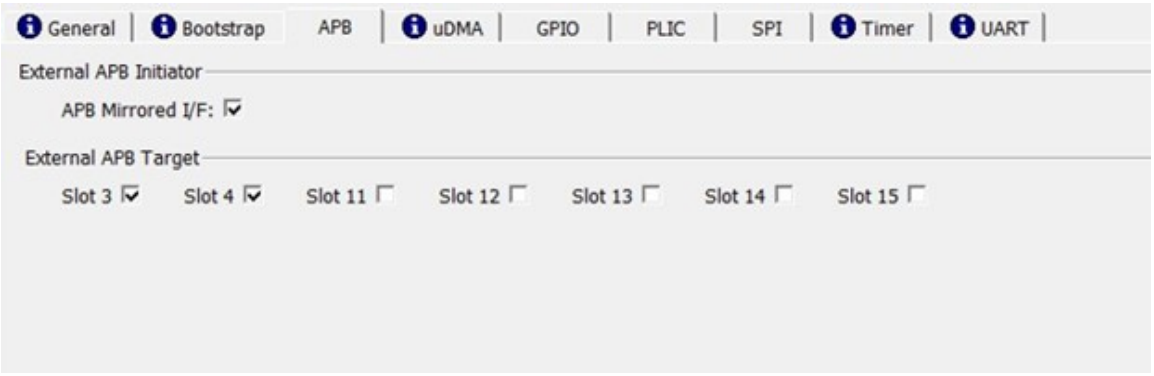
1. From the **Catalog**, drag the MIV_ESS IP core to Smart Design.
2. In the **Create Component** dialog box, enter MIV_ESS_C0 as the component name, and click **OK**.
3. In the MiV ESS Configurator screen, perform the following configurations:
   - Navigate to **General** tab, and make sure that the configurations are same as shown in the following figure.

     Figure 1-8. General Tab

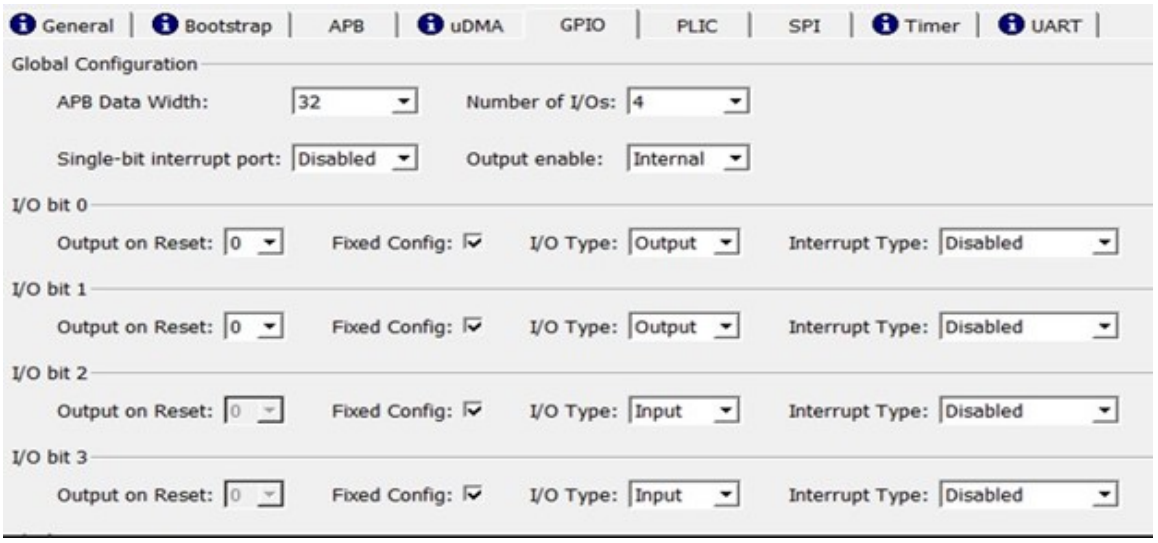
   - Navigate to **APB** tab, and select APB configuration as shown in the following figure.

     Figure 1-9. APB Tab

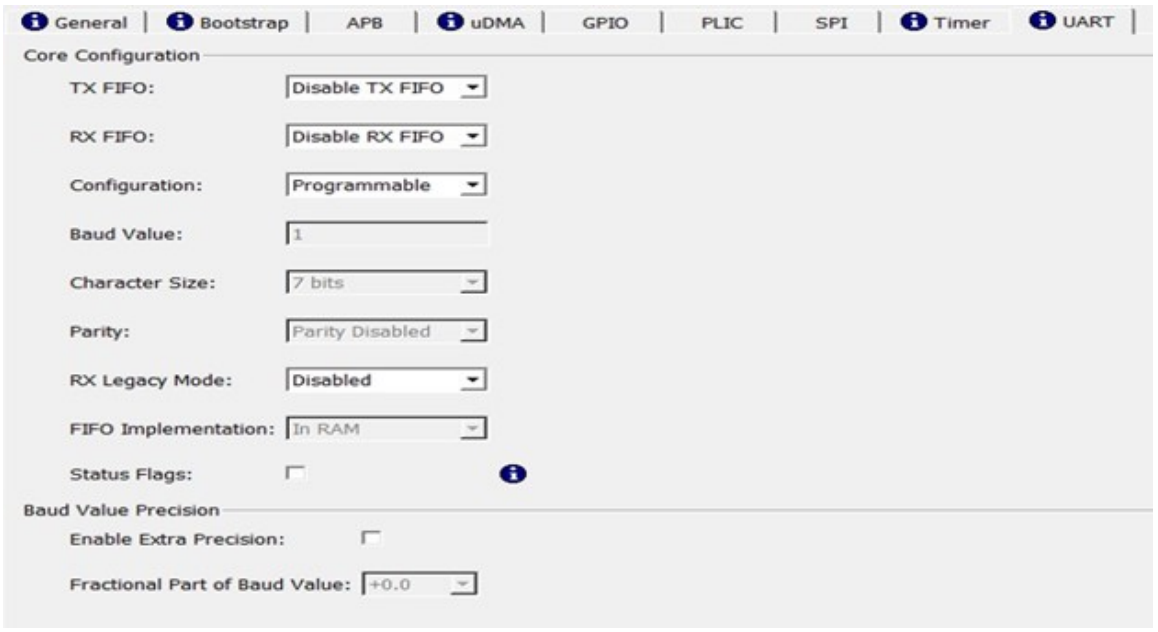
   - Navigate to **GPIO** tab, and make sure that the configurations are same as shown in the following figure.

Figure 1-10. GPIO Tab



– Navigate to **UART** tab, and make sure that the configurations are same as shown in the following figure.

Figure 1-11. UART Tab

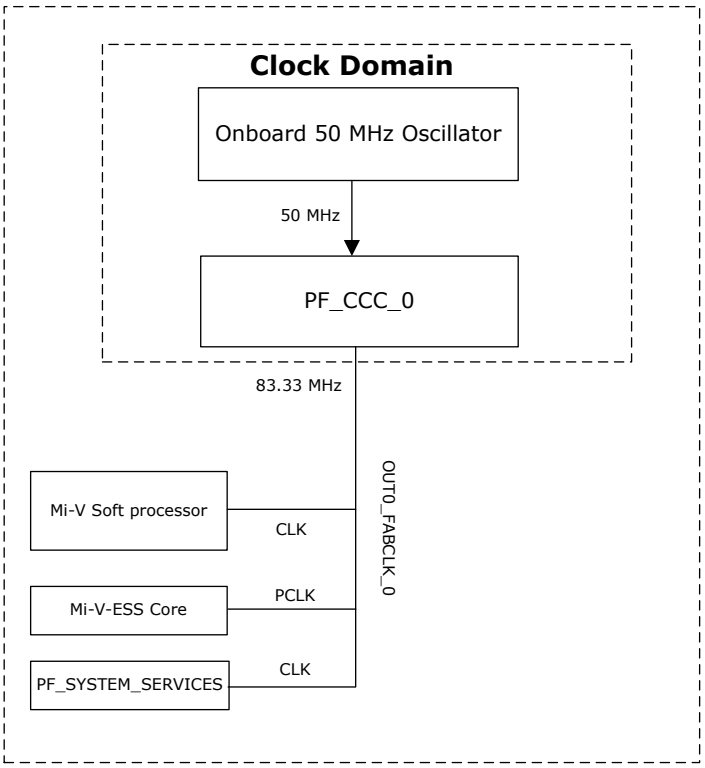


APB module used in the MIV_ESS core. ABP interface is used to connect Base peripherals. The final addresses for PF_SYSTEM_SERVICES, CoreGPIO, and CoreUARTapb targets are translated into 0x6300_0000, 0x6500_0000, and 0x6100_0000 respectively.

1.5 Clocking Structure (Ask a Question)

The following figure shows the clocking structure of the demo design. The design uses 83.33 MHz frequency.

Figure 1-12. Clocking Structure



2. Libero Design Flow (Ask a Question)

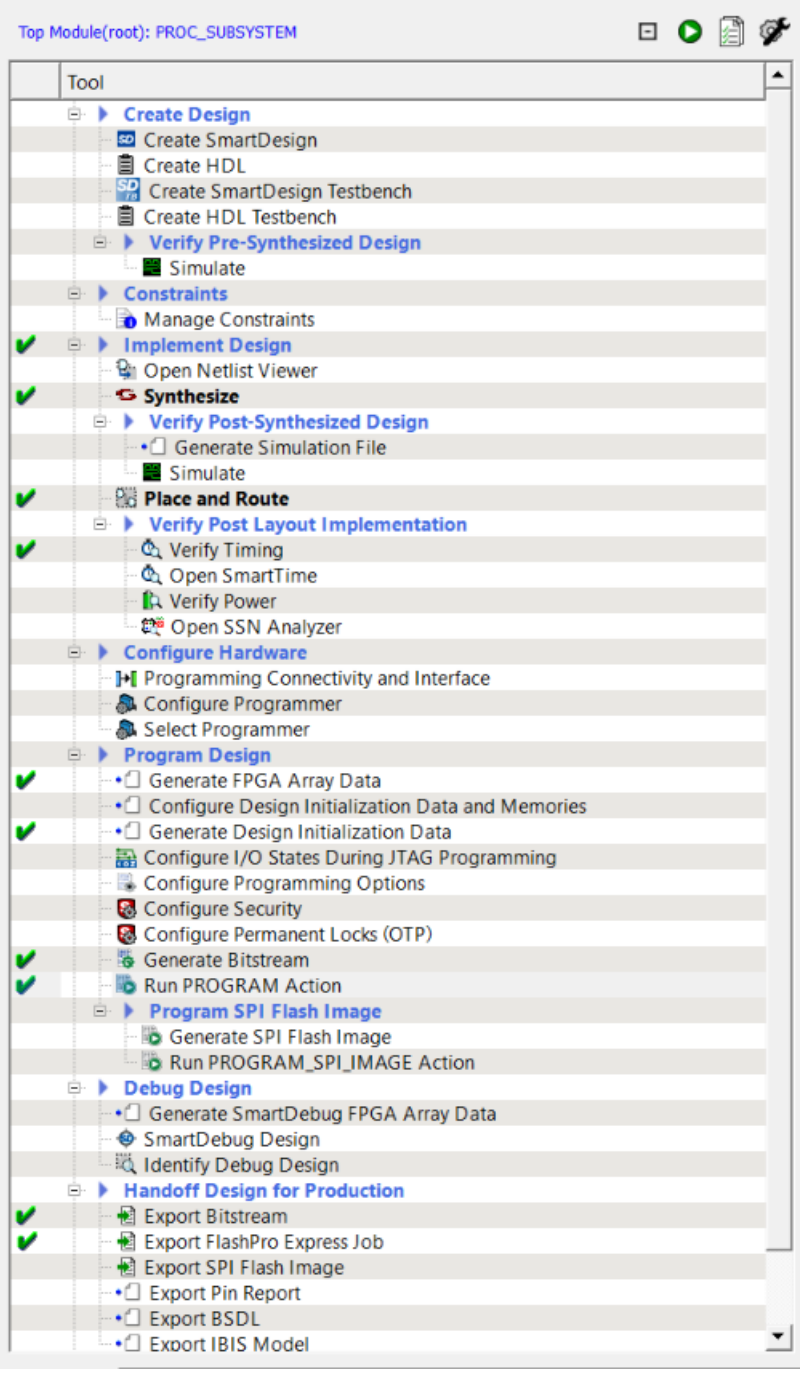
The Libero design flow involves running the following processes in the Libero SoC PolarFire:

- 2.1. Synthesize
- 2.2. Place and Route
- 2.3. Verify Timing
- 2.4. Generate FPGA Array Data
- 2.5. Configure Design Initialization Data and Memories
- 2.6. Configure Programming Options
- 2.7. Generate Bitstream
- 2.8. Run PROGRAM Action
- 2.9. Export FlashPro Express Job

Important: To initialize the TCM in PolarFire using the system controller, a local parameter **l_cfg_hard_tcm0_en**, in the `miv_rv32_subsys_pkg.v` file should be changed to 1'b1 prior to synthesis. See the MIV_RV32 Handbook. This user guide can be downloaded from the Libero SoC Catalog.

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

Figure 2-1. Libero Design Flow Options



2.1 Synthesize (Ask a Question)

To synthesize the design, perform the following steps:

1. On the **Design Flow** window, double-click **Synthesize**.
   When the synthesis is successful, a green tick mark appears, see Figure 2-1.
2. Right click **Synthesize** and select **View Report** to view the synthesis report and log files in the **Reports** tab.

Important: The `top.srr` and `top_compile_netlist.log` files are recommended to be viewed for debugging synthesis and compile errors.

2.2 Place and Route (Ask a Question)

The Place and Route process requires the I/O, timing, and floor planner constraints. The demo design includes following constraint files in the Constraint Manager window:

- The `io_constraints.pdc` file for the I/O assignments
- The `timing_user_constraints.sdc` file for timing constraints

To Place and Route, on the **Design Flow** window, double click **Place and Route**.

When place and route is successful, a green tick mark appears next to Place and Route.

Important: The file `top_place_and_route_constraint_coverage.xml` is recommended to be viewed for place and route constraint coverage.

2.3 Verify Timing (Ask a Question)

To verify timing, perform the following steps:

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

2.4 Generate FPGA Array Data (Ask a Question)

To generate the FPGA array data, perform the following steps:

1. On the **Design Flow** window, double click **Generate FPGA Array Data**.
2. A green tick mark is displayed after the successful generation of the FPGA array data, see Figure 2-1.

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

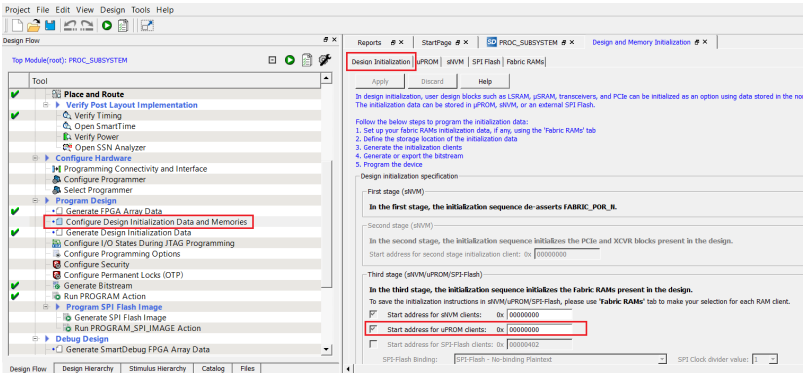
The **Configure Design Initialization Data and Memories** step generates the TCM initialization client and adds it to sNVM, µPROM, or an external SPI flash, based on the type of nonvolatile memory selected. In the demo, the TCM initialization client is stored in the µPROM.

This process requires the user application executable file (hex file) to initialize the TCM blocks on device power-up. The hex file (`application.hex`) is available in the `DesignFiles_Directory\TCL_Scripts\src\softconsole` folder. When the hex file is imported, a memory initialization client is generated for TCM blocks.

Follow these steps:

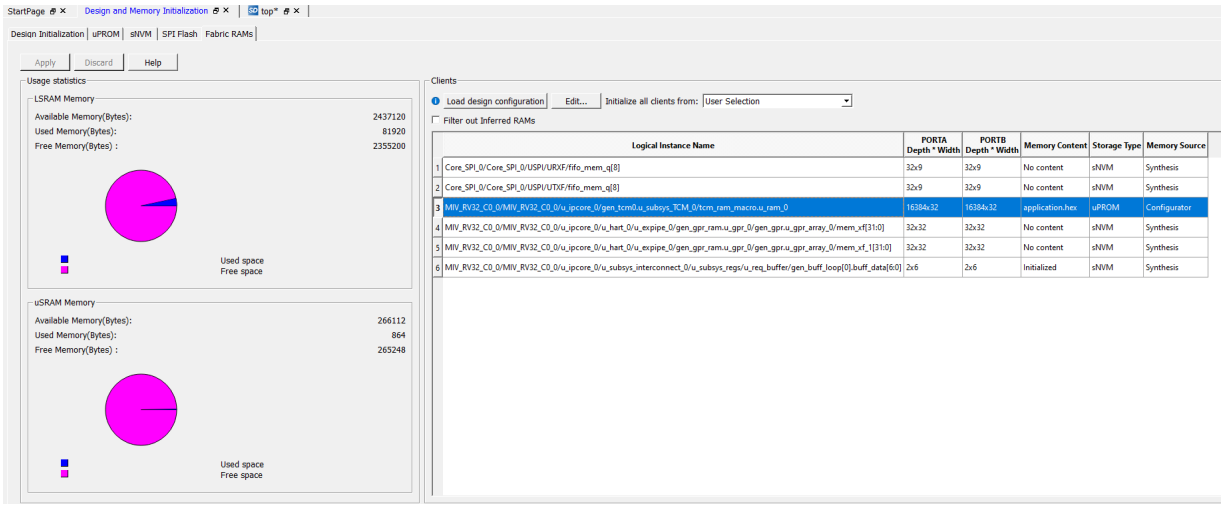
1. On the **Design Flow** window, double click **Configure Design Initialization Data and Memories**. The **Design and Memory Initialization** window opens, see the following figure.

Figure 2-2. Design and Memory Initialization



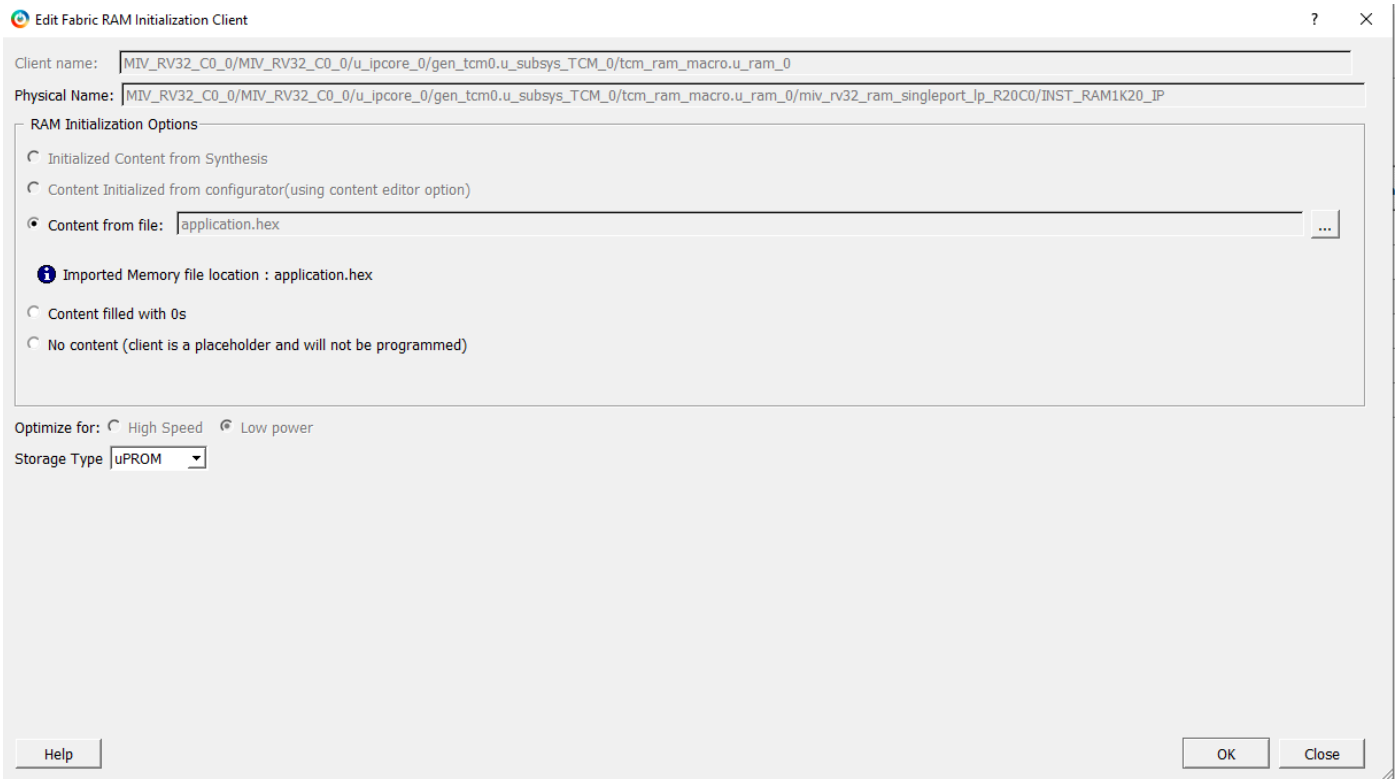
2. Select the **Fabric RAMs** tab, select the **tcm_ram** client from the list, and click **Edit**, see the following figure.

Figure 2-3. Fabric RAMs Tab



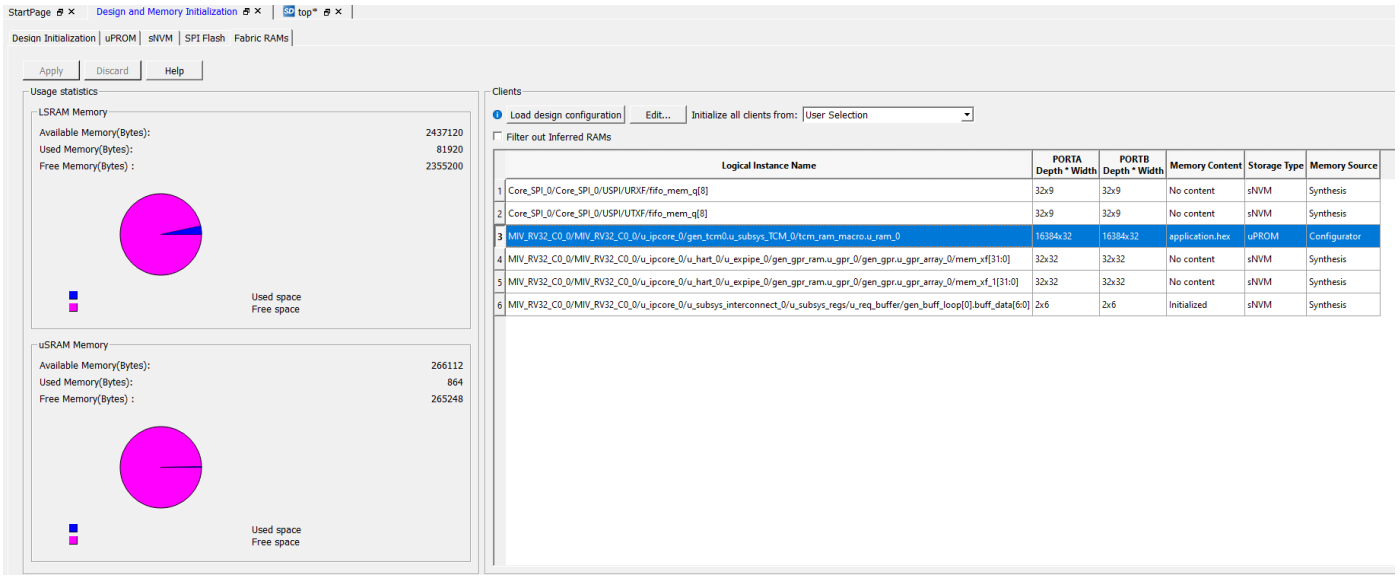
3. In the **Edit Fabric RAM Initialization Client** dialog box, select the **Content from file** option, and locate the `application.hex` file from `DesignFiles_directory\TCL_Scripts\src\softconsole` folder and Click **OK**, see the following figure.

Figure 2-4. Edit Fabric RAM Initialization Client



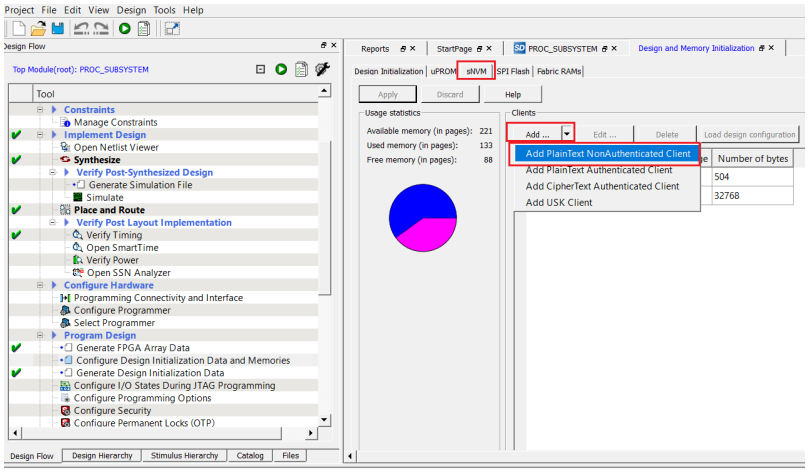
4. Click **Apply**, see the following figure.

Figure 2-5. Apply Fabric RAM Content



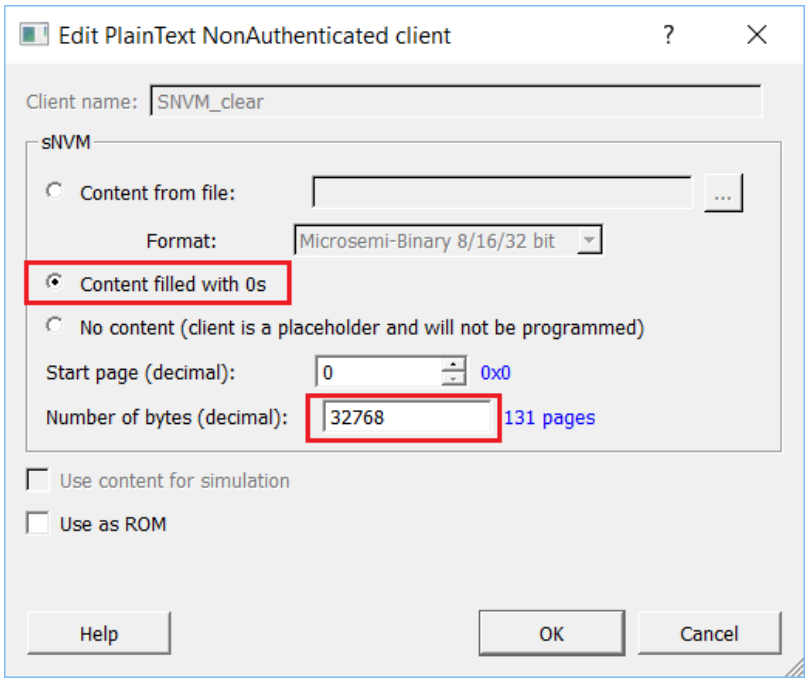
5. **Select sNVM** tab -> Select **Add** from the list -> click **Add PlainText NonAuthenticated Client**, see the following figure.

Figure 2-6. Add PlainText NonAuthenticated Client Option



6. In the preceding step, select a client and click **Edit**.
7. In the **Edit PlainText NonAuthenticated client** dialog box, select **Content filled with 0s** option and provide the **Number of bytes**. Click **OK**.

Figure 2-7. Edit PlainText NonAuthenticated Client



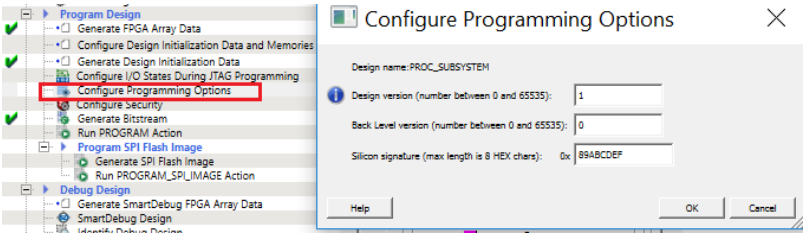
8. In the **Design Initialization** tab, click **Apply**.
9. On the **Design Flow** window, click **Generate Initialization Data** to generate design initialization data.

After successful generation of the Initialization data, a green tick mark appears next to **Generate Initialization Data** option, see the Figure 2-1.

2.6 Configure Programming Options (Ask a Question)

The Design version and user code (Silicon signature) are configured in this step. Double click **Configure Programming** Options to give values, see the following figure.

Figure 2-8. Configure Programming Options



2.7 Generate Bitstream (Ask a Question)

To generate the bitstream, perform the following steps:

1. On the **Design Flow** window, double click **Generate Bitstream**. When the bitstream is successfully generated, a green tick mark appears.
2. Right click **Generate Bitstream** and select **View Report** to view the corresponding log file in the **Reports** tab.

2.8 Run PROGRAM Action (Ask a Question)

After generating the bitstream, the PolarFire device must be programmed with the system services design.

To program the PolarFire device, perform the following steps:

1. Ensure that the following jumper settings are set on the evaluation board.

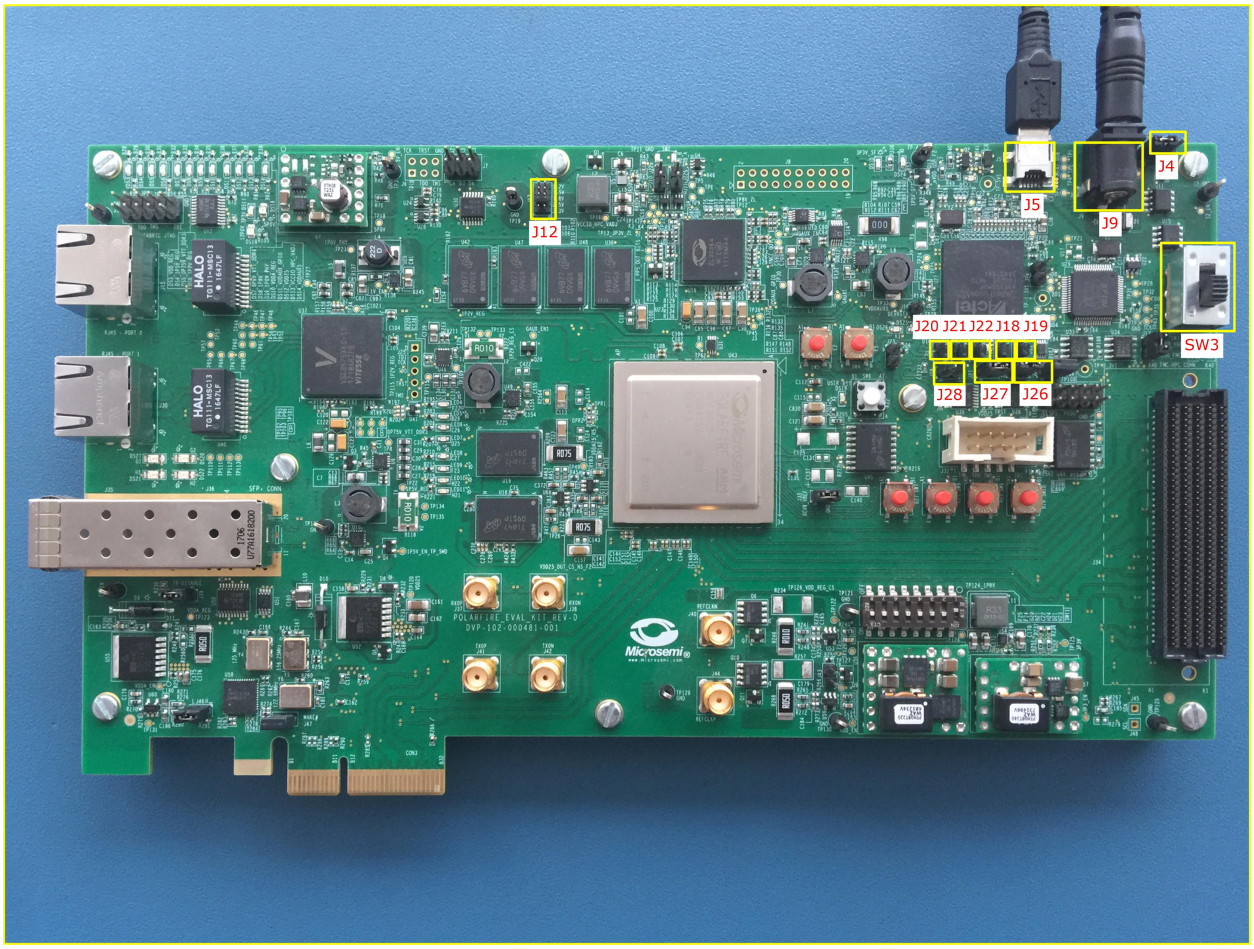
Table 2-1. Jumper Settings for PolarFire Device Programming—Evaluation Kit

Jumper	Description
J18, J19, J20, J21, and J22	Close pin 2 and 3 for programming the PolarFire FPGA through FTDI
J28	Close pin 1 and 2 for programming through the on-boardFlashPro5
J4	Close pin 1 and 2 for manual power switching using SW3
J12	Close pin 3 and 4 for 2.5 V

2. Connect the power supply cable to the connector J9 on Evaluation board.
3. Connect the USB cable from the host PC to the J5 on Evaluation board (FTDI port).
4. Power on the board using the slide switch SW3 on Evaluation board.

The following figure shows the board setup of evaluation kit.

Figure 2-9. Board Setup—Evaluation Kit



The device is successfully programmed and the onboard LEDs 7, 8, 9, 10, and 11 glow. A green tick mark appears next to Run PROGRAM Action.

2.9 Export FlashPro Express Job (Ask a Question)

On the Design Flow window, double click **Export FlashPro Express Job**. When the job file is successfully generated, a green tick mark appears, see Figure 2-1

On the **Design Flow** window, double click **Run PROGRAM Action**.

The device is successfully programmed and the onboard LEDs 4, 5, 6, 7, and 8 glow. A green tick mark appears next to **Run PROGRAM Action**.

To program the device using the `.job` file provided along with the demo design, see 5. Appendix 1: Programming the Device Using FlashPro Express.

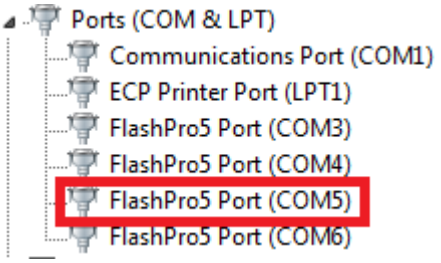
3. Setting up the Serial Terminal Program - PuTTY (Ask a Question)

The user application receives system service commands on the serial terminal through the UART interface. This chapter describes how to set up the serial terminal program.

To setup PuTTY, perform the following steps:

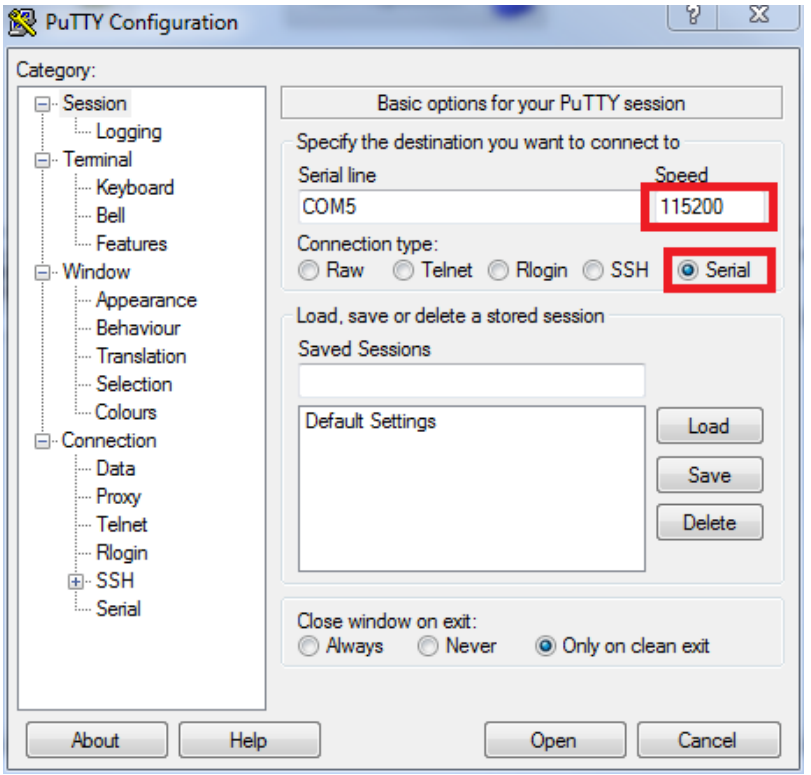
1. Connect the USB cable from the host PC to the J5 (USB) port on the Evaluation board.
2. Connect the power supply cable to the J9 connector on the Evaluation board.
3. Power on the board using the SW3 slide switch for Evaluation board.
4. From the host PC, click **Start** and open **Device Manager** to note the second highest COM Port number and use that in the PuTTY configuration. In this example, COM Port 5 (**COM5**) is selected, see the following figure. COM Port-numbers may vary.

Figure 3-1. Finding the COM Port



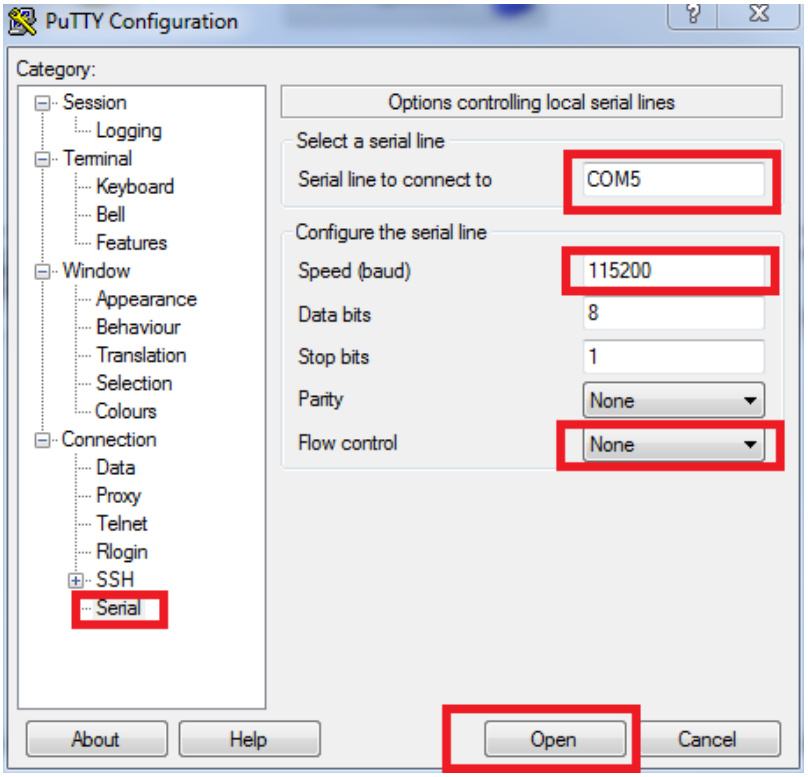
5. On the host PC, click **Start**, find and select PuTTY program.
6. Select **Serial** as the **Connection type**, see the following figure.

Figure 3-2. Select Serial as the Connection Type



7. Set the **Serial line to connect** to COM port number noted in step 4.
8. Set the **Speed (baud)** to 115200, see the following figure.
9. Set the **Flow control** to **None**, see the following figure and click **Open**.

Figure 3-3. PuTTY Configuration



PuTTY opens successfully and this completes the serial terminal emulation program setup. See 4. Running the Demo.

4. Running the Demo (Ask a Question)

This chapter describes how to run the system services demo using the serial terminal program (PuTTY). The prerequisite for the following procedure is to program the device and to set up the serial terminal. For more information on setting up the serial terminal, see 3. Setting up the Serial Terminal Program - PuTTY.

To run the demo, perform the following steps:

1. Power on the board using the SW3 slide switch.
   System services options are displayed on the PuTTY, see the following figure.

Figure 4-1. System Services Options

```
 **** PolarFire Device and Design system services Example ****
Notes: Return data from System controller is displayed byte-wise with LSB first
   Input data is provided LSB first. Each ASCII character is one Nibble of data

Select Service:
1. Read Device Serial number
2. Read Device User-code
3. Read Device Design-info
4. Read Device certificate
5. Read Digest
6. Query security
7. Read debug information
8. Digital signature
9. Secure NVM services
a. PUF Emulation
b. Nonce
```

2. Enter 1 to select **Read Device Serial number**.
   The 128-bit Device Serial Number (DSN) is displayed, see the following figure.

Figure 4-2. Device Serial Number

```
------------------------------------------------------------------------
Device serial number:  278398138C7F3CDBC1A4AB3059CECAB5
------------------------------------------------------------------------
```

Each PolarFire FPGA device has a unique, publicly readable, 128-bit DSN. The DSN can be used in cryptographic protocols to uniquely identify the device. The DSN comprises two 64-bit fields: FSN and SNM.

Factory Serial Number (FSN)—The first (most significant) field is the FSN. It is a pseudo-random per-device unique value assigned during Microchip's manufacturing test and persists for the lifetime of the device.

Serial Number Modifier (SNM)—The second component is the SNM. It is initialized during the factory test and is destroyed during the recoverable zeroization action. If the device is subsequently recovered, a new SNM is assigned such that each SNM generated for a given FSN, is unique.

Important: The system services main menu is displayed after the execution of any of the options.

3. Enter 2 to select **Read Device User-code**.
   The 32-bit device USERCODE/Silicon signature is displayed, see the following figure.

Figure 4-3. Device User-code

```
------------------------------------------------------------------------
32bit USERCODE/Silicon signature (MSB first):  89ABCDEF
------------------------------------------------------------------------
```

This can be configured from `Design flow->Program and Debug Design->Configure Programming Options`.

4. Enter 3 to select **Read Device Design-info**.
   See Figure 4-4, the device design information consists of:
   - 256-bit user-defined Design ID
   - 16-bit design version
     This can be configured from `Design flow->Program and Debug Design->Configure Programming Options`. In auto update programming, the current design version is compared with the available images in external SPI flash to initiate the auto update on power up.
   - 16-bit design back-level
     This can be configured from `Design flow->Program and Debug Design->Configure Security`. When back level protection is enabled, the device can only be programmed if the target design version is more than the back level value.

   Figure 4-4. Device Design Information

   ```
   ------------------------------------------------------------------------
   Design ID:
    6F44746F700000000000000000000000
    00000000000000000000000000000000
   Design Version:  0100
   Design Back-Level:  0000
   ------------------------------------------------------------------------
   ```

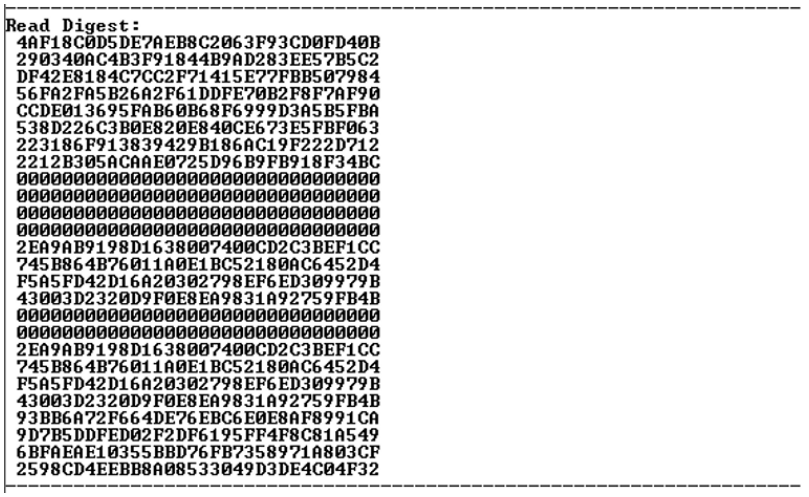
5. Enter 4 to select **Read Device Certificate**.
   The device supply chain assurance certificate is displayed, see Figure 4-5.
   For more information about decoding the device certificate, see 6. Appendix 2: Device Certificate Information.

Figure 4-5. Device Certificate


```
Device Certificate:
 308203CC30820352A003020102021240
 F49CDB72092BF9C8BF0F1E063658F76E
 01300A06082A8648CE3D040303304931
 0B3009060355040613025553310C300A
 060355040B1303536F43310D300B0603
 55040A13044D53434331 1D301B060355
 04031314316233646233316236303130
 3131353530373765302017 0D31353130
 31343030303734375A180F3939393931
 3233313233353935395A304E314C300B
 060355042C1304303030303013060355
 04040C0C506F6C61724669726520 2020
 302806035504 2A0C214D504633303054
 20202020202020202020202020202020
 20202020202020202020307630100607
 2A8648CE3D020106052B810400220362
 0004F62000124EFF538E5438F5EDFC19
 9154F091E120FA35BA3E80898BF48C95
 FFF56B2EC1365B0D1BC7D00968DEB0D5
 7384DA7C85F9107D9049B67EDEA76694
 5216264B3F934E7D9FC4101D92102CD9
 5FA574C94BCD14938B68B37CB1B8980D
 3F3A810900409AEF658B6D9900821100
 278398138C7F3CDBC1A4AB3059CECAB5
 A38201D6308201D23012060A2B060104
 0182BD64010104040000000030818F06
 0A2B0601040182BD640102048180FFFF
 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
 FFFFFFFFFFFFFFFFFFFFFFFFFF003012
 060A2B0601040182BD6401030404FFFF
 FFFF308186060A2B0601040182BD6401
 04047800000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 000000000000000000000030818C060A
 2B0601040182BD640105047E00000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000300A06082A86
 48CE3D0403030368003065023100A9EC
 5E497493635C4EEECDC55D5D27078A65
 D2E53DC64321A0AA61F156659FBFA9C3
 359B22772AF0B527B9B634EF99BB0230
 0E9FD2343049DA9E1D8DB4EAF65E0E2F
 4A635BC0DECBCFEA3F5D57BE461E7642
 BFFBBA3E9B0B82C08A851BCD41A285A8
 00000000000000000000000000000000
 00000000000000000000000000000000
 00000000000000000000000000000000
```

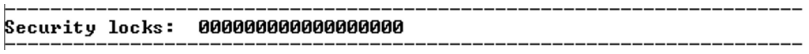

6. Enter 5 to select **Read Digest**.
   The 416 byte Digest contains the fabric digest, sNVM digest, and user key digests. The Digest protects data integrity. The following figure shows the 416 byte digest displayed. For more information about decoding the digest output, see 9. Appendix 5: Digest Information

Figure 4-6. Digest



7. Enter 6 to select **Query Security**.
   The non-volatile states of user security locks are displayed, see the following figure.

Figure 4-7. Security Locks Information



The design does not include any security settings for device safety. Security locks can be configured from `Design flow->Program and Debug Design->Configure Security`. For more information about security locks, see 6. Appendix 2: Device Certificate Information.

> **Important:** If security locks are not handled properly, the device can go into the locked state.

8. Enter 7 to select **Read debug information**.
   The debug information is displayed, see the following figure.

Figure 4-8. Debug Information

In Figure 4-8, the highlighted 4 bytes E7000000 (LSB first) indicates the number of times (in this example,0x000000E7), the device was programmed (programming cycles). For more information about the Debug Info fields, see 6. Appendix 2: Device Certificate Information.

9. Enter 8 to select **Digital Signature**.
   The following figure displays digital signature in both Raw and DER formats.

Figure 4-9. Digital Signature

```
------------------------------------------------------------------------
Digital Signature service:
48 byte hash value:
 911601143D01D7E2676B0434FF45D7BB
 5634E5CBF8BB6F685E5A2D7D038AB2F9
 911601143D01D7E2676B0434FF45D7BB
Raw format:
Digital Signature service successful.
Output Digital Signature - Raw format:
 2B04CA57A62DE3F6F9E910EBBE4BBD03
 89F5DB43244AC2BBC51A4475ED95B007
 73F5BB94828064851B7A6F1E08A33E31
 AC3D3985F185FECD392CA68552F2199E
 8F74C00B198ECF2ACD2378637F94F458
 51FF5AF30983757CDC1F6123DE4699AD
DER format:

Digital Signature service successful.
Output Digital Signature - DER format:
 3066023100A8ED918D910DB652980E4D
 EB43CFD8BE29EF631B0D7421D57C72BA
 F7E66AFF29D2F939D6C621BCE069AFCB
 BAF31A1DF20231009F6A900B7511A4AE
 84C72C61D11E3F0002F7EEF6C3BA7C82
 7AF86FD3F83389C1005AD627AD99326C
 21A8A7A589F1D978
------------------------------------------------------------------------
```

The digital signature service takes a user-supplied SHA384 hash and signs it with the devices private key. The application randomly generates the SHA384 hash value. The Digital Signature service sends the hash value to the System Controller. The Athena core runs the Elliptic Curve Digital Signature Algorithm (ECDSA) using the hash and the device private key to generate the signature.

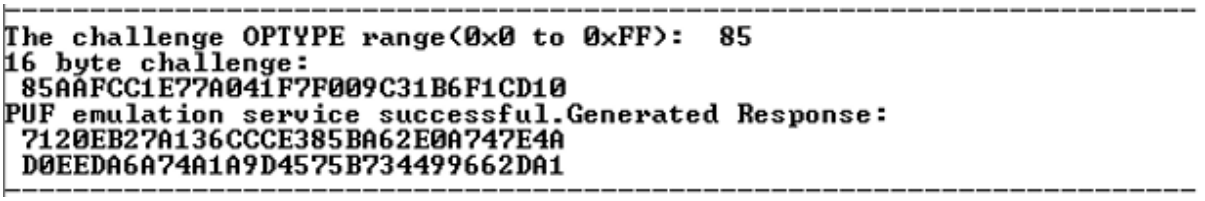
10. Enter 9 to select **Secure NVM**.
    When the sNVM page/ module address is entered (in this case, 0), the randomly generated 60 byte data is written to the specified sNVM page and read back, see Figure 4-10.

Figure 4-10. Secure NVM Services

```
------------------------------------------------------------------------
Secure NVM (sNVM) Functions
sNVM write format
    Non-authenticated plain-text

------------------------------------------------------------------------------
Enter the sNVM page/module address (1 HEX byte.Page value range is from 0x01 to
0x83) and Press Enter key:
1 1

Input Data(60 Byte):
554D9036446F48C97E9A95C1124B9747
E55E699A554D9036446F48C97E9A95C1
124B9747E55E699A554D9036446F48C9
7E9A95C1124B9747E55E699A
Secure NVM write successful.
------------------------------------------------------------------------

Data read from sNVM region:
554D9036446F48C97E9A95C1124B9747
E55E699A554D9036446F48C97E9A95C1
124B9747E55E699A554D9036446F48C9
7E9A95C1124B9747E55E699A00000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00100060020000000000000000000000
00000000000000000000000000000000
00000000000000000000006003000000
56501820802E2528AF397294554D9036
446F48C97E9A95C1124B9747E55E699A
0102D42D01944FC11400E77F00000000
00012FA50000000072200000 1E020000
00DE013695FAB60B00000000
```

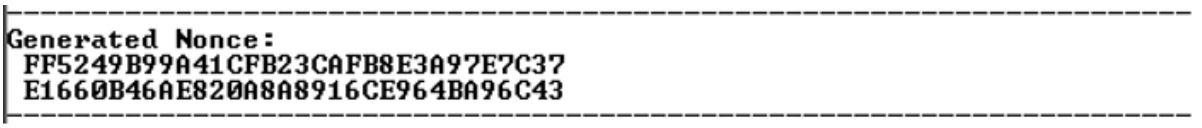
11. Enter ‘a’ (without quotes) to select **PUF Emulation_service**
    The PUF emulation service provides a mechanism for authenticating a device, or for generating a pseudo random bit strings that can be used for different purposes. When this service is selected, the service by default accepts a 128-bit challenge and an 8-bit optype, and returns a 256-bit response unique to the challenge and the optype, see the following figure.

Figure 4-11. PUF Emulation Service



12. Enter ‘b’ to select **Nonce service**.
    The following figure displays 32 byte nonce value. The nonce service provides the ability to strengthen the Deterministic Random Bit Generator (DRBG) of the Athena by providing an alternate entropy source to use as additional seed data in its DRBG functions.

Figure 4-12. Generated Nonce



5. Appendix 1: Programming the Device Using FlashPro Express (Ask a Question)

This chapter describes how to program the PolarFire device with the Job programming file using a FlashPro programmer. The default location of the Job file are located at following location:

```
mpf_an4663_v2024p1_df\Programming_Job\top.job
```

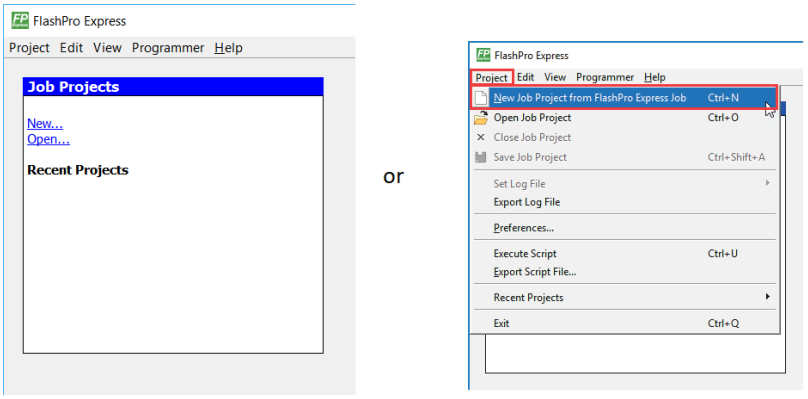
To program the PolarFire device using FlashPro Express, perform the following steps:

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

> **Important:** The power supply switch must be switched off while making the jumper connections.

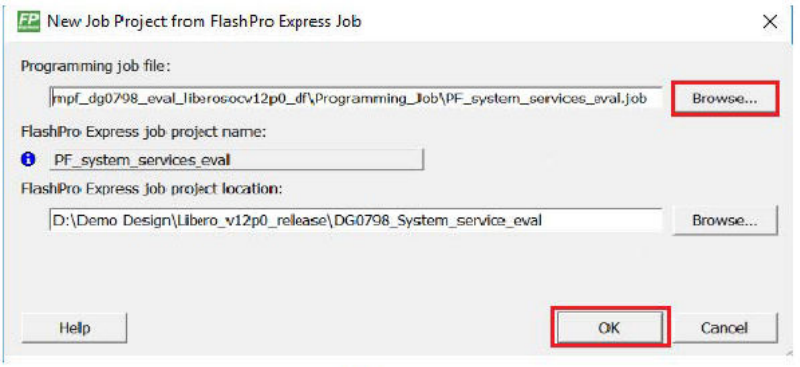
2. Connect the power supply cable to the J9 connector on the board.
3. Connect the USB cable from the Host PC to the J5 (FTDI port) on the board.
4. Power on the board using the SW3 slide switch.
5. On the host PC, launch the FlashPro Express software.
6. To create a new job project, click **New** or
   In the **Project** menu, select **New Job Project from FlashPro Express Job**, see the following figure.

Figure 5-1. FlashPro Express Job Project



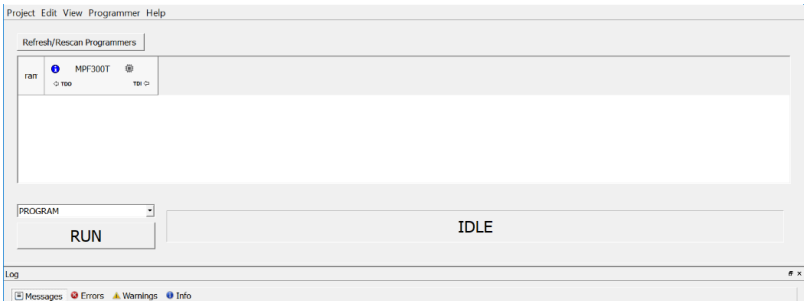
7. Enter the following in the **New Job Project from FlashPro Express Job** dialog box:
   - **Programming job file**: Click **Browse**, and navigate to the location where the `.job` file is located and select the file. The default location is: *<download_folder>mpf_an4663_v2024p1_df\Programming_Job*.
   - **FlashPro Express job project location**: Click **Browse** and navigate to the location where you want to save the project.

Figure 5-2. New Job Project from FlashPro Express Job



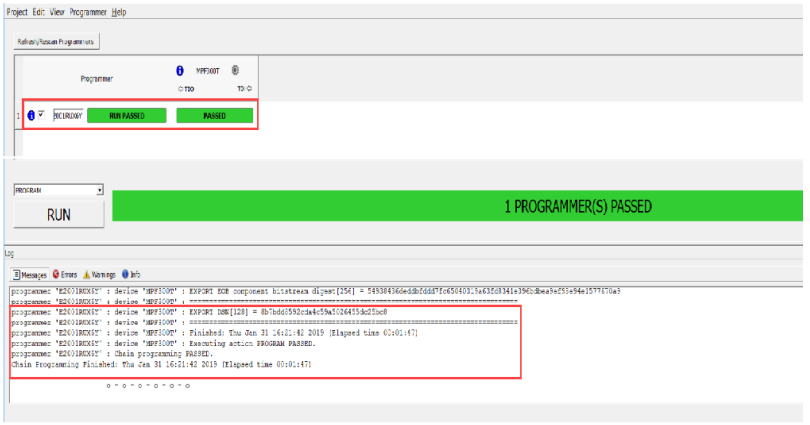
8. Click **OK**. The required programming file is selected and ready to be programmed in the device.
9. The following figure displays the FlashPro Express window. Confirm that a programmer number appears in the Programmer field. If it does not, confirm the board connections and click **Refresh/ Rescan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

Figure 5-3. Programming the Device



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

Figure 5-4. FlashPro Express—RUN PASSED



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

6. Appendix 2: Device Certificate Information (Ask a Question)

The Device Certificate is a 1024 byte Microchip-signed X-509 certificate programmed during manufacturing. The certificate is used to guarantee the authenticity of a device and its characteristics.

The following table lists the main fields of the device certificate.

Table 6-1. Device Certificate Fields (1024 bytes)

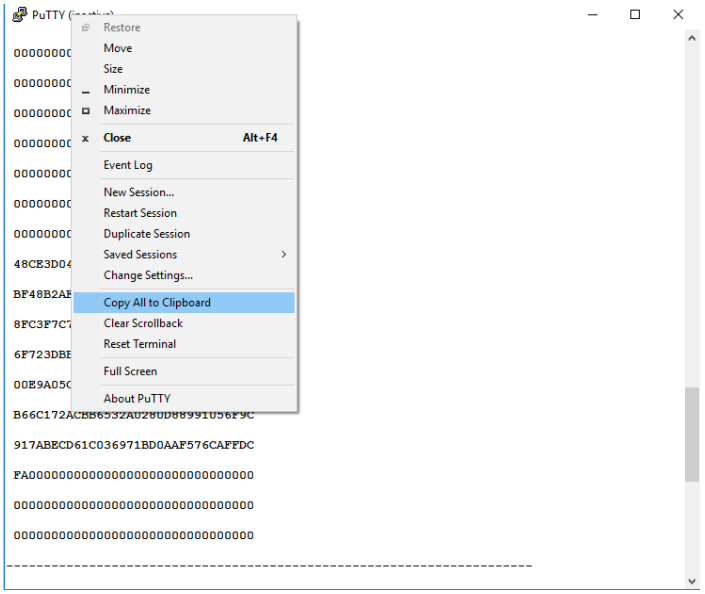
Offset (Byte)	Length (Bytes)	Data
4	854	Signed region of certificate
234	120	Device Public Key
368	16	DSN

The device certificate is encoded in the ASN.1 format. To view the content, the certificate must be decoded to a user readable format using the online JAVA tool: http://lapo.it/asn1js/#.

For decoding a certificate, perform the following steps:

1. Right click **PuTTy**, select **Copy All to Clipboard**, and paste the same to notepad, see the following figure.

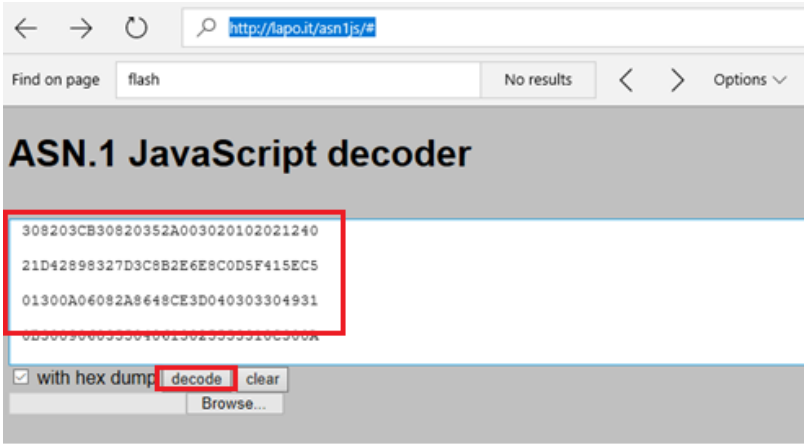
Figure 6-1. Copy Device Certificate



2. Copy the 1024 bytes of device certificate from notepad to ASN.1 decoder, see the following figure and click **decode** button.

Important: The sample device certificate (1024 bytes) is provided at:
For Evaluation kit
`mpf_an4663_v2024p1_df\Device_Certificate\sample.txt`

Figure 6-2. Certificate Decoding Using Java Script



3. The web page displays all the fields in certificate, see the following figure.

Figure 6-3. Decoded Certificate

7. Appendix 3: Query Security (Ask a Question)

The following table lists each security lock bit and its features.

Table 7-1. Security Locks Fields

Byte	Bit	Lock	Description
0			
	0	UL_DEBUG	Debug instructions disabled
	1	UL_SNVM_DEBUG	sNVM debug disabled
	2	UL_LIVEPROBE	Live probes disabled
	3	UJTAG_DISABLE	User JTAG interface disabled
	4	JTAG_BS_DISABLE	JTAG boundary scan disabled
	5	UL_TVS_MONITOR	External access to system Temperature and Voltage Sensor (TVS) disabled
	6	JTAG_MONITOR	JTAG fabric monitor enabled
	7	JTAG_TAP	JTAG TAP disabled
1			
	0	UL_PLAINTEXT	Plain text passcode unlock disabled
	1	UL_FAB_PROTECT	Fabric erase/write disabled
	2	UL_EXT_DIGEST	External digest check disable
	3	UL_VERSION	Replay protection enabled
	4	UL_FACT_UNLOCK	Factory test disabled
	5	UL_IAP	IAP disabled
	6	UL_EXT_ZEROIZE	External zeroization disabled
	7	SPI_SLAVE_DISABLE	SPI port disabled
2-8		Reserved	Reserved

8. Appendix 4: Debug Information (Ask a Question)

The following table lists the debug information bit fields.

Table 8-1. Debug Information Fields

Byte Offset	Size (Bytes)	Parameter	Description
0	32	RESERVED	
32	4	TOOL_INFO	• Reflects the TOOL_INFO passed in during ISC_ENABLE prior to programming • IAP sets this to 0
36	1	TOOL_TYPE	Tool type used to program device: 1 = JTAG, 2 = IAP, and 3 = SPI_SLAVE
37	4	RESERVED	—
41	7	RESERVED	—
48	1	UIC_STATUS	Design initialization status
49	1	UIC_SOURCE_TYPE	Design initialization Data source type when execution finished or halted
50	2	RESERVED	
52	4	UIC_START_ADDRESS	Design initialization Data source address when execution finished or halted
56	4	UIC_INSTR_ADDRESS	Design initialization Data instruction count from the start of Design initialization execution
60	4	CYCLECOUNT	Programming cycle count
64	1	IAP_ERROR_CODE	IAP error information
65	7	RESERVED	—
72	4	IAP_LOCATION	External SPI flash memory address that was used during IAP

9. Appendix 5: Digest Information (Ask a Question)

The following table lists the digest information bit fields.

Table 9-1. Digest Information Bit Fields

Offset (byte)	Size (bytes)	Value	Note
0	32	CFD	Fabric digest
32	32	CCDIGEST	Fabric Configuration segment digest
64	32	SNVMDIGEST	sNVM Digest
96	32	ULDIGEST	User lock segment
128	32	UKDIGEST0	User Key Digest 0 in User Key segment (includes SRAM PUF activation code and device Integrity bit)
160	32	UKDIGEST1	User Key Digest 1 in User Key segment
192	32	UKDIGEST2	User Key Digest 2 in User Key segment (UPK1)
224	32	UKDIGEST3	User Key Digest 3 in User Key segment (UEK1)
256	32	UKDIGEST4	User Key Digest 4 in User Key segment (DPK)
288	32	UKDIGEST5	User Key Digest 5 in User Key segment (UPK2)
320	32	UKDIGEST6	User Key Digest 6 in User Key segment (UEK2)
352	32	UPDIGEST	User Permanent lock (UPERM) segment
384	32	FDIGEST	Digest for Factory Key Segments
Total	416 bytes		

10. Appendix 6: Running the TCL Script (Ask a Question)

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

To run the TCL, follow the steps below:

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

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

For more information about TCL scripts, refer to `mpf_an4663_v2024p1_df/TCL_Scripts/TCL_Script_readme.txt`

Contact Technical Support for any queries encountered when running the TCL script.

11. Appendix 7: References (Ask a Question)

This section lists documents that provide more information about system services and other IP cores used to build the system services.

- For more information on Design and Data Security Services, see PolarFire Family Security User Guide.
- For more information about the CoreJTAGDEBUG IP core, see *CoreJTAGDebug_HB.pdf*. This user guide is downloaded from the Libero SoC Catalog.
- For more information about the MIV_RV32 IP core, see *MIV_RV32 Handbook*. This user guide is downloaded from the Libero SoC Catalog.
- For more information about the PolarFire initialization monitor, see PolarFire FPGA and PolarFire SoC FPGA Power-up and Resets User Guide.
- For more information about how to build a Mi-V processor subsystem for PolarFire devices, see AN4997: PolarFire FPGA Building a Mi-V Processor Subsystem (Earlier TU0775).
- For more information about the PF_CCC IP core, see PolarFire Family Clocking Resources User Guide.
- For more information about Libero, ModelSim, and Synplify, see Libero SoC Documentation.

12. Revision History (Ask a Question)

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

Revision	Date	Description
B	07/2024	The following is a summary of the changes made in this revision: • Replaced the reference to Firmware Catalog with the driver example in GitHub in 1.1. PF_SYSTEM_SERVICES Core Overview. • Updated the document for Libero® SoC v2024.1 • Updated package information in Table 1-3 • Updated Figure 1-2 • Updated Figure 1-3 • Removed "The Mi-V processor supports up to 120 MHz" from 1.4.1.1. PF_CCC_0 Configuration section • Updated Figure 1-5 • Updated Figure 1-6 • Updated Figure 1-7 • Added 1.4.1.3. MIV ESS Core • Updated Figure 1-12 • Updated Figure 2-3 • Updated Figure 2-4 • Updated Figure 2-5 • Updated 11. Appendix 7: References
A	07/2022	The following is a summary of the changes made in this revision. • The document was migrated to the Microchip template. • The document number was updated to DS00004663A from 50200798. • The document ID was updated to AN4663 from DG0798. • Removed all SPLASH Board related content from document. • Updated Figure 1-3. • Updated Figure 1-5. • Updated Figure 5-2.
8.0	—	The following is a summary of the changes made in this revision. • Replaced text CoreSysServices_PF with PF_SYSTEM_SERVICES. • Updated Figure 1-1, Figure 1-2, and Figure 1-12. • Replaced Figure 1-3 and Figure 4-1 through Figure 4-11.
7.0	—	Added 10. Appendix 6: Running the TCL Script.
6.0	—	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.
5.0	—	Updated the document for Libero SoC v12.0.
4.0	—	The following is a summary of the changes made in this revision. • Updated for Libero SoC PolarFire® v2.3. • Merged SPLASH kit related content.
3.0	—	Updated the document for Libero SoC PolarFire v2.2.
2.0	—	Updated the document for Libero SoC PolarFire v2.1.
1.0	—	The first publication of this document.

Microchip FPGA Support

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

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

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

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

Microchip Information

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: 978-1-6683-4968-7

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	**Australia - Sydney** Tel: 61-2-9868-6733	**India - Bangalore** Tel: 91-80-3090-4444	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393
Atlanta Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455	**China - Beijing** Tel: 86-10-8569-7000	**India - New Delhi** Tel: 91-11-4160-8631	**Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829
Austin, TX Tel: 512-257-3370	**China - Chengdu** Tel: 86-28-8665-5511	**India - Pune** Tel: 91-20-4121-0141	**Finland - Espoo** Tel: 358-9-4520-820
Boston Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088	**China - Chongqing** Tel: 86-23-8980-9588	**Japan - Osaka** Tel: 81-6-6152-7160	**France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79
Chicago Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075	**China - Dongguan** Tel: 86-769-8702-9880	**Japan - Tokyo** Tel: 81-3-6880- 3770	**Germany - Garching** Tel: 49-8931-9700
Dallas Addison, TX Tel: 972-818-7423 Fax: 972-818-2924	**China - Guangzhou** Tel: 86-20-8755-8029	**Korea - Daegu** Tel: 82-53-744-4301	**Germany - Haan** Tel: 49-2129-3766400
Detroit Novi, MI Tel: 248-848-4000	**China - Hangzhou** Tel: 86-571-8792-8115	**Korea - Seoul** Tel: 82-2-554-7200	**Germany - Heilbronn** Tel: 49-7131-72400
Houston, TX Tel: 281-894-5983	**China - Hong Kong SAR** Tel: 852-2943-5100	**Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906	**Germany - Karlsruhe** Tel: 49-721-625370
Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380	**China - Nanjing** Tel: 86-25-8473-2460	**Malaysia - Penang** Tel: 60-4-227-8870	**Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44
Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800	**China - Qingdao** Tel: 86-532-8502-7355	**Philippines - Manila** Tel: 63-2-634-9065	**Germany - Rosenheim** Tel: 49-8031-354-560
Raleigh, NC Tel: 919-844-7510	**China - Shanghai** Tel: 86-21-3326-8000	**Singapore** Tel: 65-6334-8870	**Israel - Hod Hasharon** Tel: 972-9-775-5100
New York, NY Tel: 631-435-6000	**China - Shenyang** Tel: 86-24-2334-2829	**Taiwan - Hsin Chu** Tel: 886-3-577-8366	**Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781
San Jose, CA Tel: 408-735-9110 Tel: 408-436-4270	**China - Shenzhen** Tel: 86-755-8864-2200	**Taiwan - Kaohsiung** Tel: 886-7-213-7830	**Italy - Padova** Tel: 39-049-7625286
Canada - Toronto Tel: 905-695-1980 Fax: 905-695-2078	**China - Suzhou** Tel: 86-186-6233-1526	**Taiwan - Taipei** Tel: 886-2-2508-8600	**Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340
	China - Wuhan Tel: 86-27-5980-5300	**Thailand - Bangkok** Tel: 66-2-694-1351	**Norway - Trondheim** Tel: 47-72884388
	China - Xian Tel: 86-29-8833-7252	**Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Poland - Warsaw** Tel: 48-22-3325737
	China - Xiamen Tel: 86-592-2388138		**Romania - Bucharest** Tel: 40-21-407-87-50
	China - Zhuhai Tel: 86-756-3210040		**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