AN4593

PolarFire FPGA How to Perform On-Demand Digest Check

Introduction

PolarFire® devices have a built-in self-test mechanism that can be used (optionally) to check the design integrity and security of the device automatically upon power-up or on-demand. The contents of all the non-volatile configuration memory segments, including security keys, security settings, and the FPGA fabric configuration, as well as sNVM memory pages declared as ROM by the user (all the write-protected pages), are tested using the digest check feature. This test provides assurance against both natural and maliciously induced failures.

Table of Contents

1. How to Perform On-Demand Digest Check

In the factory and user security segment, each logical page contains an automatically generated digest, calculated dynamically when programming the data to be written. The digest includes an overall value covering the data to be programmed for the FPGA fabric. Also, the digests are calculated and stored for the sNVM pages marked as ROM. The digests can be verified on-demand by the user, either internally using a system service or externally using a programming instruction. In addition, the user can automatically run digest checks on each power-up.

An endurance limit specifies how often a digest of the FPGA fabric can be run. For more information about the FPGA configuration memory endurance limits, see PolarFire FPGA Datasheet. Therefore, depending on how the system is deployed and used (for example, how often it is powered up), the on-demand digest check is more appropriate for testing the integrity of the FPGA fabric.

1.1 On-Demand Digest Check

The on-demand digest check recalculates and compares digests of selected non-volatile memories (FPGA fabric, sNVM, and security segments) with the stored digests. A failure of any digest check results in the tamper event being triggered for a user action. The on-demand digest check is invoked by calling a digest check design system service.

If the digest check is performed on the FPGA fabric, then the FPGA fabric is automatically placed in the suspended state before commencing the digest check operation. Except for LSRAMs, all the FPGA fabric logic elements and uSRAMs hold their current state during the suspend state. Upon completion of the fabric digest, the suspend state is automatically exited. Since the LSRAMs do not retain the user data after performing a digest check on the FPGA fabric. The status of the fabric digest check must be monitored by a state machine implemented in the fabric.

After checking the status of the fabric digest check, the state machine needs to issue a design reset or device reset, depending on the design requirements. The device reset reinitializes the device and designs memories according to the user configuration. The user design reset does not reinitialize the device or LSRAMs. The user design reset must be implemented to reset the user design. If the design uses LSRAMs initialization at power-up, then the user design needs to issue a device reset; otherwise, a design reset is sufficient to restart the design. Use RESET_DEVICE tamper response signal for device reset.

The FPGA fabric design, including LSRAMs content, is not affected when the digest check is performed on sNVM or security segments. See PolarFire FPGA and PolarFire SoC FPGA Security User Guide for more information on Digest Check Service.

This application note provides a design example of how to perform an on-demand digest check.

1.2 Design Requirements

The following table lists the hardware and software required to perform a fabric digest check operation.

Table 1-1. Resource Requirements

Requirement	Version
Host PC Operating system	Windows® 10
Hardware	
PolarFire® Evaluation Kit with MPF300TS-1FCG1152 device	Rev D
Software	
Libero SoC	**Note:** See the `readme.txt` file provided in the design files for the software versions used with this reference design.
Tera Term	

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

1.3 Prerequisites

Download the design files from the following location:

www.microchip.com/en-us/application-notes/AN4593

Download and install Libero SoC (as indicated in the website for this design) on the host PC from the following location:
Libero SoC Documentation

- **Important:** The provided design example is created for MPF300TS-1FCG1152 device (Non-ES device).
- **Important:** To create Libero project using TCL script, refer 2. Appendix: Running the TCL Script.

1.4 Design Description

The design example provided with the application note performs the following operations:

1. Initiates the system service request to perform the following system services:
   - Fetch Device Serial Number (DSN)
   - Fetch USERCODE
   - Digest Check on FPGA Fabric
2. Demonstrates how to perform design reset or device reset after a fabric digest check

   .
3. Provides user interface through UART terminal

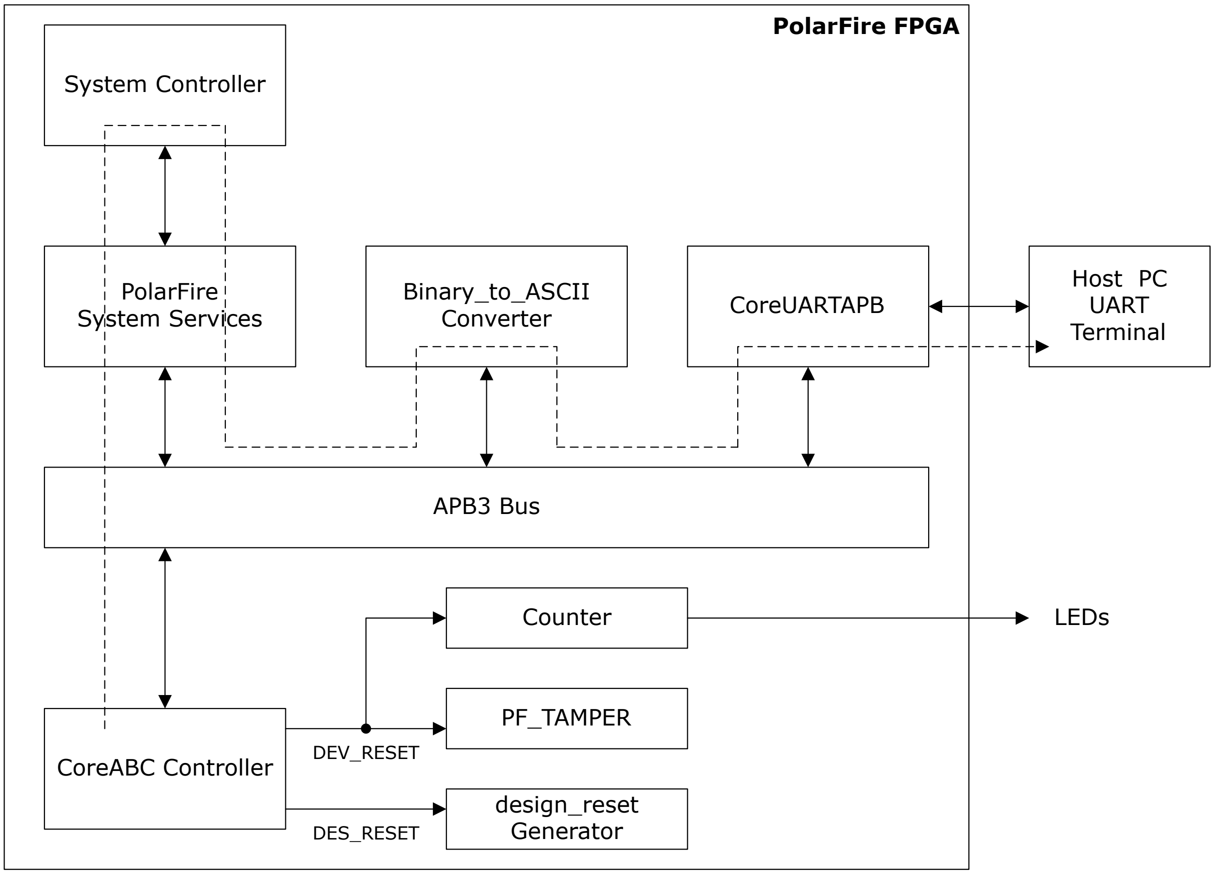
   .

System services are system controller actions initiated from the user design. PF_SYSTEM_SERVICES IP must be used to initiate system service requests. PF_SYSTEM_SERVICES IP provides a set of registers, which are accessible through the APB interface for executing the system services. On the other hand, the PF_SYSTEM_SERVICES IP interfaces to the PolarFire System Controller through a system service interface. The System Controller has a 2 KB mailbox RAM for additional input parameters and outputs from the service.

A CoreABC processor is used as a controller to perform the preceding operations. CoreABC (APB Bus Controller) is a simple, configurable, low gate count, programmable state machine/controller primarily targeted towards implementing of AMBA APB-based designs. For information about the instructions supported by the CoreABC, see CoreABC handbook.

The following figure shows the block diagram of the example design. The design is implemented without consuming the LSRAMs since the LSRAMs content is not retained after performing the fabric digest check.

Figure 1-1. Design Block Diagram



The dotted line in Figure 1-1 shows the data flow. The data flow shown in Figure 1-1 is explained as follows:

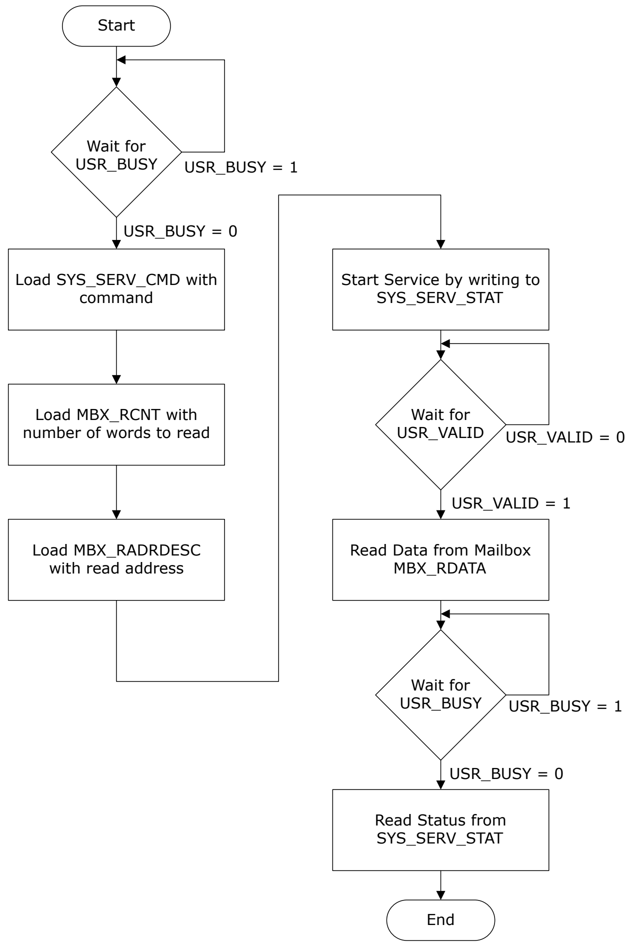
1. CoreABC Controller initiates a system service through PF_SYSTEM_SERVICES IP.
2. System Controller performs the requested action and writes the result to the System Controller mailbox.
3. CoreABC processor reads the result and converts it into ASCII format using the Binary_to_ASCII converter.
4. CoreABC processor prints the result on a host PC UART terminal through CoreUARTAPB controller.
5. After digest check, the CoreABC processor issues a device reset or a design reset based on user input. The PF_TAMPER macro is used to issue a device reset.

The System Services are invoked by writing to appropriate PF_SYSTEM_SERVICES IP registers with the system service command and any additional information required to perform the system service. Then the PF_SYSTEM_SERVICES IP initiates necessary transactions on the system service interface. The PF_SYSTEM_SERVICES IP asserts the USR_BUSY signal when the IP is busy executing the service. Upon completion of a system service, a status code is written to the status register, and the USR_BUSY signal gets de-asserted. For services requiring the mailbox reads, the signal is de-asserted only when the required number of reads for the service is completed. If a new service request is initiated when the USR_BUSY is in progress, the new request is ignored until the current service completes. For more information about PF_SYSTEM_SERVICES IP, see UG0848 PolarFire System Services User Guide.

The following flowchart shows the sequence of instructions coded in the CoreABC program to read DSN or USERCODE using System Services. The command value and number of words (32 bits) to read (MBX_RCNT) are service dependent:

- For read DSN service, the command is 0x00 and MBX_RCNT is 4
- For read USERCODE service, the command is 0x01 and MBX_RCNT is 1

Figure 1-2. Sequence of Instructions to Read DSN or USERCODE



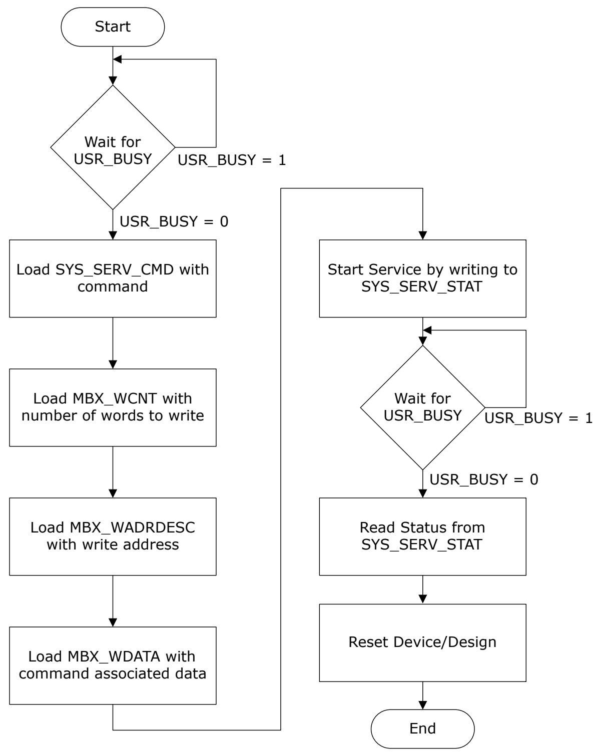
The command value to perform the digest check service is 0x47, and this service request requires additional information (OPTIONS [15:0], as listed in the following table) to specify the area for which the digest check must be performed. The additional information is provided through mailbox writes. In this reference design, 0x1 is written to the mailbox (MBX_WDATA) to perform a digest check on the FPGA fabric. Users can change this parameter to perform digest on other NVM components. The digest check service does not expect any return data other than the status of the service.

Table 1-2. OPTIONS [15:0]

OPTIONS [i]	Name	Description
0	CHECK FABRIC	Enables fabric digest
1	CC	Enables digest of fabric configuration data
2	SNVM	Enables digest of sNVM pages marked as ROM
3	UL	Enables digest of user security segment
4	UKDIGEST0	Enables digest of user key segment containing SRAMPUF data
5	UKDIGEST1	Enables digest of user key segment containing UEK (User EC key)
6	UKDIGEST2	Enables digest of user key segment containing UPK1
7	UKDIGEST3	Enables digest of user key segment containing UEK1

The following flowchart shows the sequence of instructions coded in the CoreABC program to perform a digest check on the FPGA fabric using the system service.

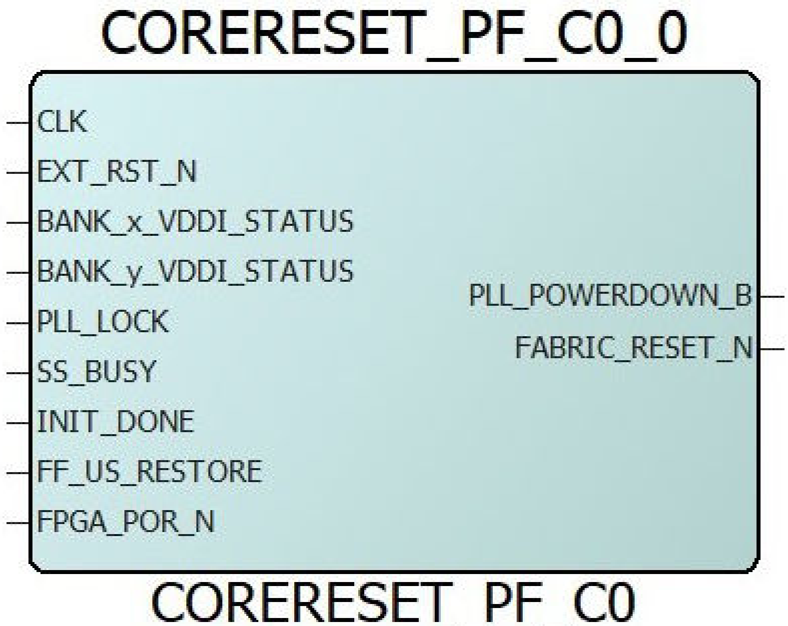
Figure 1-3. Sequence of Intructions to Perform Digest Check on FPGA Fabric



The FPGA fabric is put into a suspended state during fabric digest check calculation, as explained in 1.1. On-Demand Digest Check. During the suspend state, the clocks are gated off to low, the PLLs are powered down, and all the I/O input buffers are disabled. After digest check, the FPGA fabric automatically comes out of its suspended state and the PLLs and clocks are re-enabled. The PLLs take time to re-acquire the lock. Since the PLL lock is used in design reset generation, the PLL lock signal needs to be bypassed until the lock is re-acquired after exiting from suspend state. If not, the design gets reset because of the loss of the PLL lock.

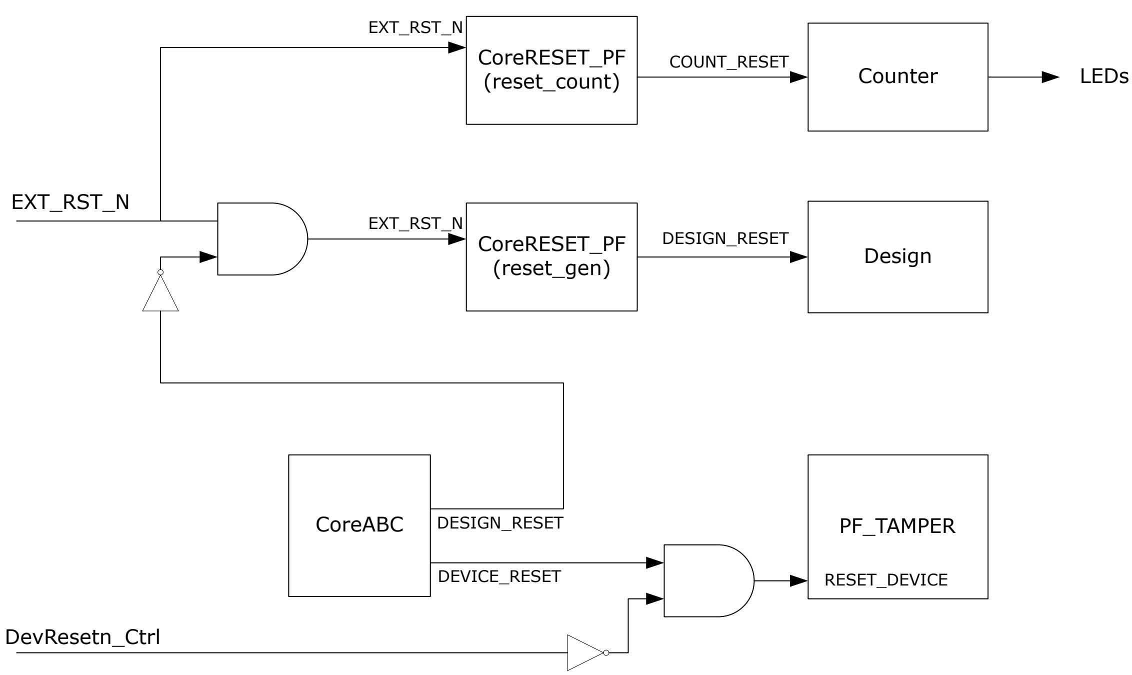
The example design uses CoreReset_PF IP core (as shown in the following figure) to serve the preceding purpose. The CoreReset_PF IP generates a reset, which is asserted asynchronously by one of the multiple potential sources and de-asserted synchronously with a specified clock. The potential sources of reset can be external reset through a GPIO, PLL lock, and INIT_DONE from PolarFire Initialization Monitor (PF_INIT_MONITOR). The CoreReset_PF IP uses USR_BUSY signals from PF_SYSTEM_SERVICES IP to bypass PLL_LOCK and EXT_RESET_N at the time of suspending state exit. This ensures that no reset occurs while the PLL is a re-acquiring lock after exiting the suspend state. FF_US_RESTORE must be tie low.

Figure 1-4. CoreReset_PolarFire Ports



If LSRAMs are used in the design, then the user design needs to issue a device reset or design reset after completing the digest check on the FPGA fabric. Based on the user input, the CoreABC processor asserts reset_device input of PF_TAMPER macro for device reset or asserts EXT_RST_N input of CoreReset_PF IP for design reset. The example design uses two CoreReset_PF IPs to show differentiation between device reset and design reset: one is used to generate a reset to complete the design except for a counter, which blinks LEDs on board, and the other IP is used to reset the counter. The counter does not get reset when the design reset is initiated from the CoreABC processor. The counter gets reset only when the device reset is issued.

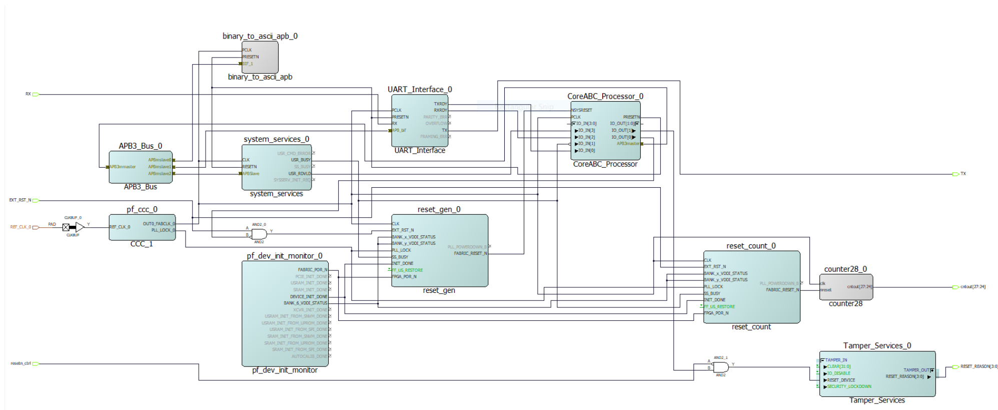
Figure 1-5. Reset Structure



Important: The RESET_DEVICE input of PF_TAMPER macro must not be promoted to top-level I/Os directly. The design example provides device reset control from an external DIP switch (SW11-DIP1) through DevResetn_Ctrl port. See Figure 1-5 for DevResetn_Ctrl port connection. SW11-DIP1 switch position must be set to ON to control the device reset from the user design. Logic '0' is driven on DevResetn_Ctrl port when the DIP1 switch position is set to ON.

The following figure shows the hardware implementation of the design example using Libero.

Figure 1-6. Hardware Implementation of the Design

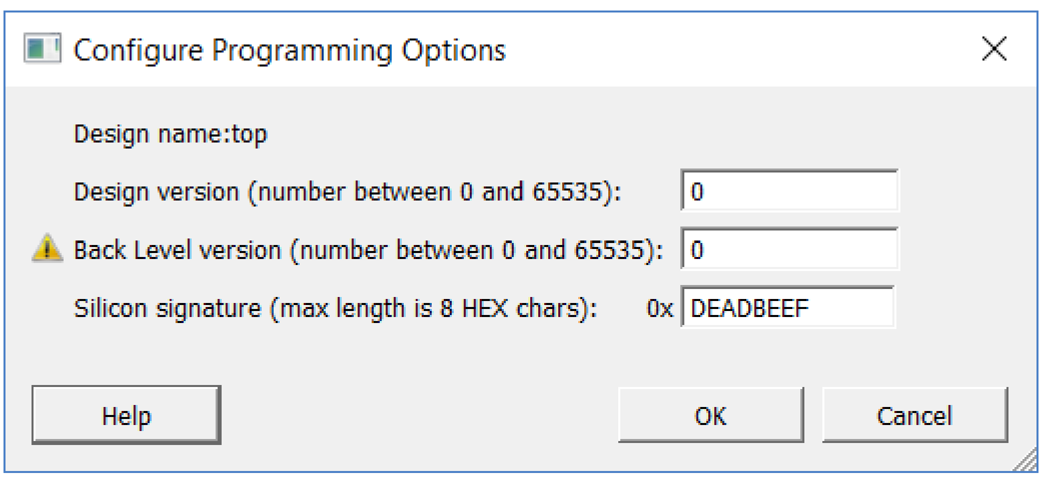


The design example also demonstrates the following system services:

- Fetch Device Serial Number system service
- Fetch JTAG USERCODE system service

The DSN is a 128-bit unique device ID set in the factory. JTAG USERCODE is a 32-bit user— configurable silicon signature to be programmed into the device. It can be set using **Program Design** > **Configure Programming Options** available in the Libero Design Flow, as shown in the following figure.

Figure 1-7. JTAG USERCODE or Sillicon Signature Configuration



1.5 How to Run the Demo

This section provides instructions to run the demo of the design example.

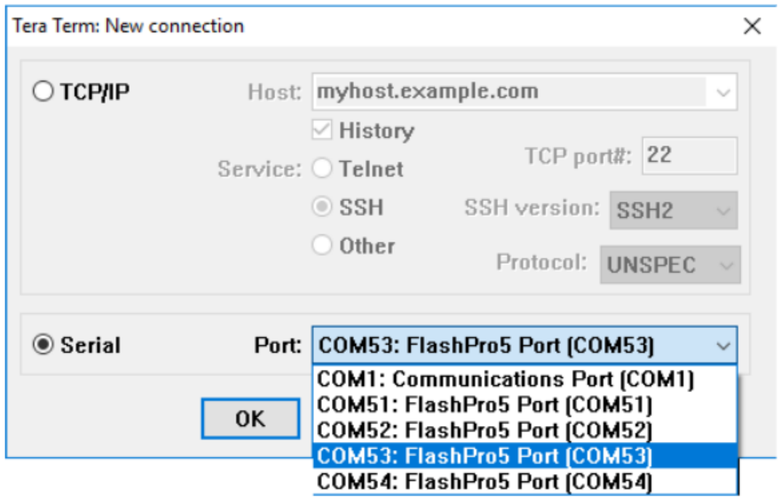
1.5.1 Tera Term Setup

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

Perform the following steps to set up the Tera Term program:

1. Ensure that the USB cable connects the host PC to the J5 (USB) port on the PolarFire Evaluation Board.
2. Start Tera Term.
3. Select **File** > **New connection…** from the Tera Term menu.
4. Select **Serial** as the **Connection type**.
5. Set the **Serial Port** to the second highest COM port number from the drop-down list, as shown in the following figure. For example, **COM53: FlashPro5 Port [COM53]** in this case.

   Figure 1-8. Select Serial Port



6. In the Tera Term window, go to **Setup** > **Serial port**, set baud rate as 115200.

This completes the Tera Term program setup.

1.5.2 Board Setup

Perform the following steps to setup the board:

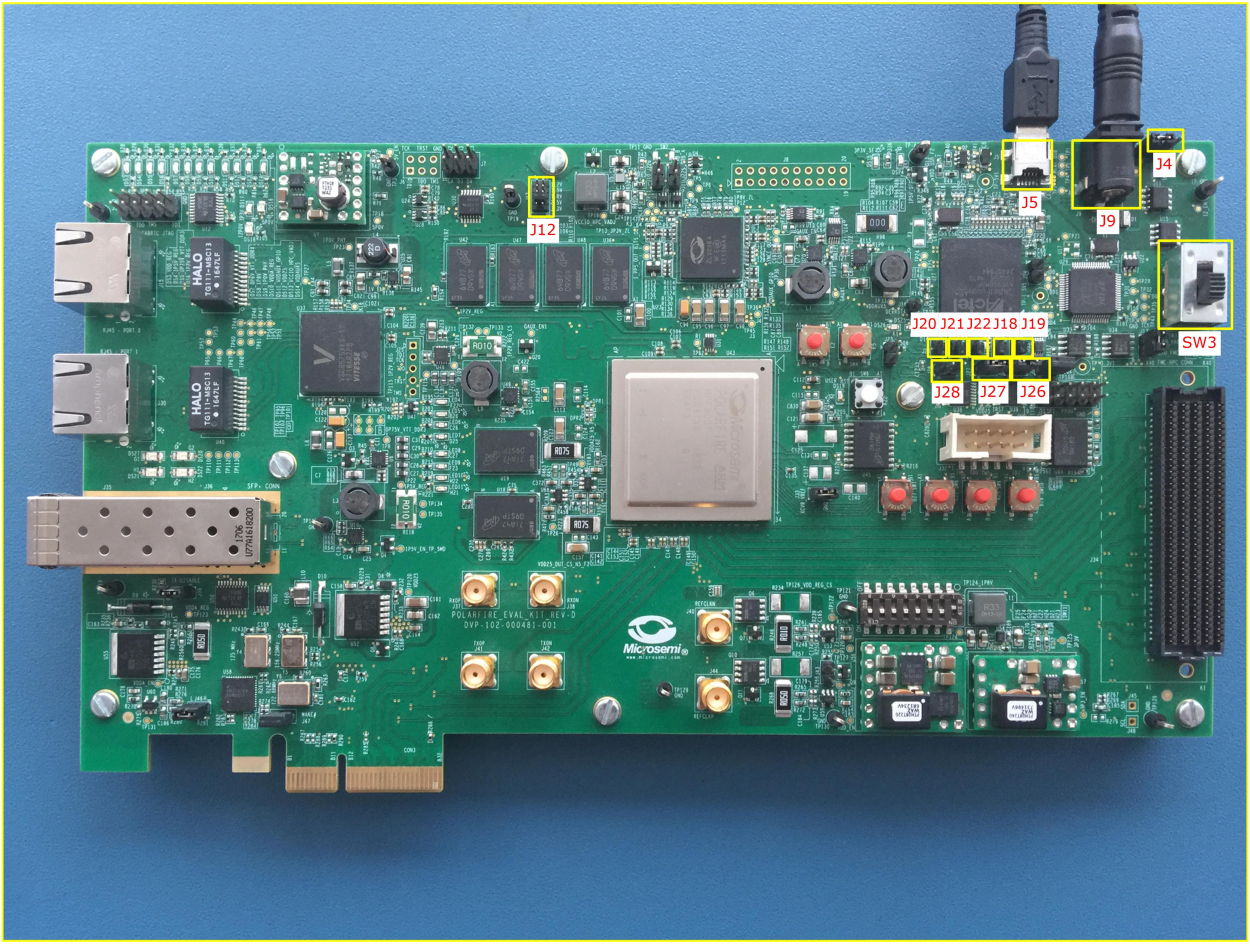
1. Ensure that the jumpers on the evaluation board are set as specified in the following table.

Table 1-3. Jumper Settings

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-board FlashPro5
J26	Close pin 1 and 2 for programming through the FTDI SPI
J27	Close pin 1 and 2 for programming through the FTDI SPI
J23	Open pin 1 and 2 for programming SPI Flash
J4	Close pin 1 and 2 for manual power switching using SW3
J12	Close pin 3 and 4 for 2.5 V
SW11	DIP1 switch position should be set to ON.

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

Figure 1-9. Board Setup



1.5.3 Program the Device

Perform the following steps to program the PolarFire device:

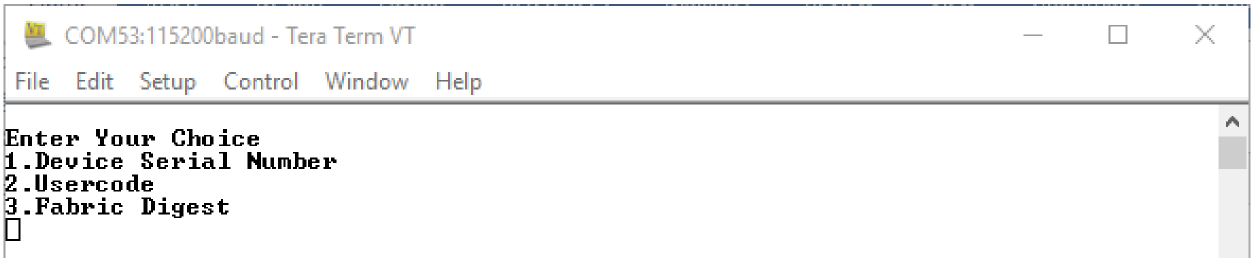
1. Open the provided Libero project.
2. Click **Run PROGRAM Action** to program the device.

Important: Users can use FlashPro Express to program the device using a job file without the need to open Libero. The programming job (`.job`) file for the demo design is available as part of the design files.

1.5.4 Running the Demo

After the device is programmed with the provided design. The design prints the menu on the Tera Term program through the UART interface, as shown in the following figure. The state of a 4-bit free-running counter is shown using four user LEDs on the board.

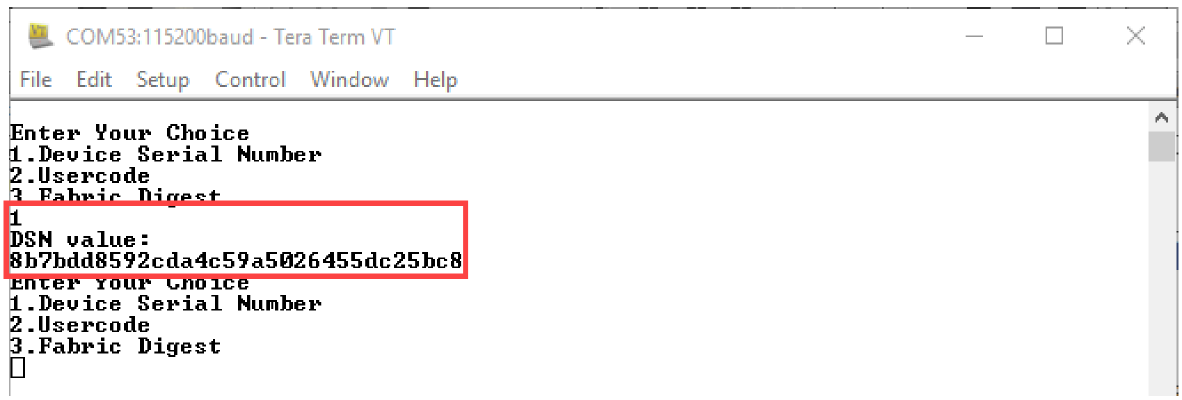
Figure 1-10. User Interface Menu



Perform the following steps to run the demo:

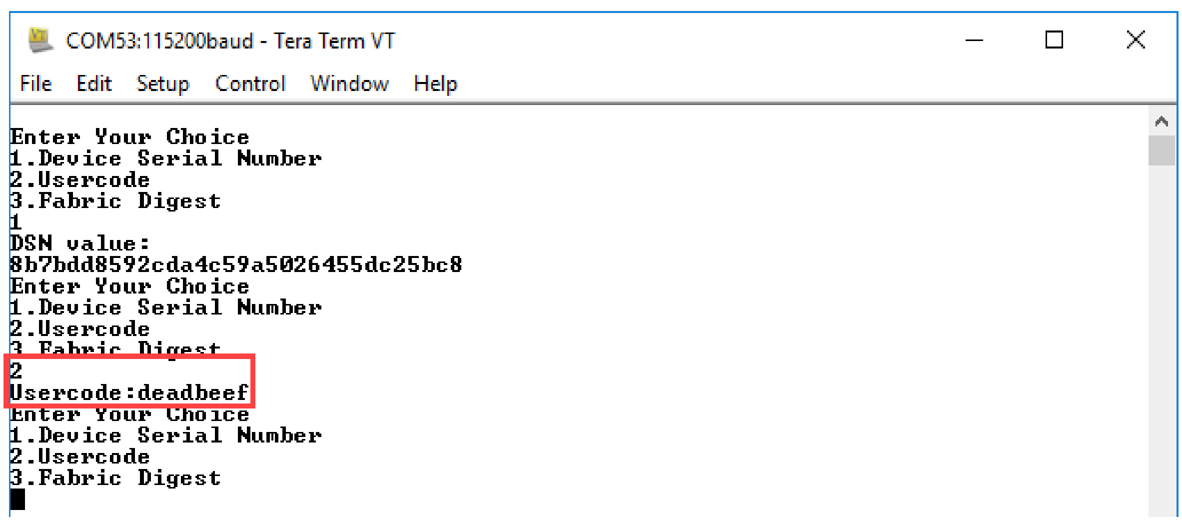
1. Press '1' to run DSN system service. As a result, the DSN of the device present on the evaluation board is printed on the terminal, as shown in the following figure. The DSN returned from the service is same as the DSN exported during device programming.

   Figure 1-11. Fetch DSN System Service



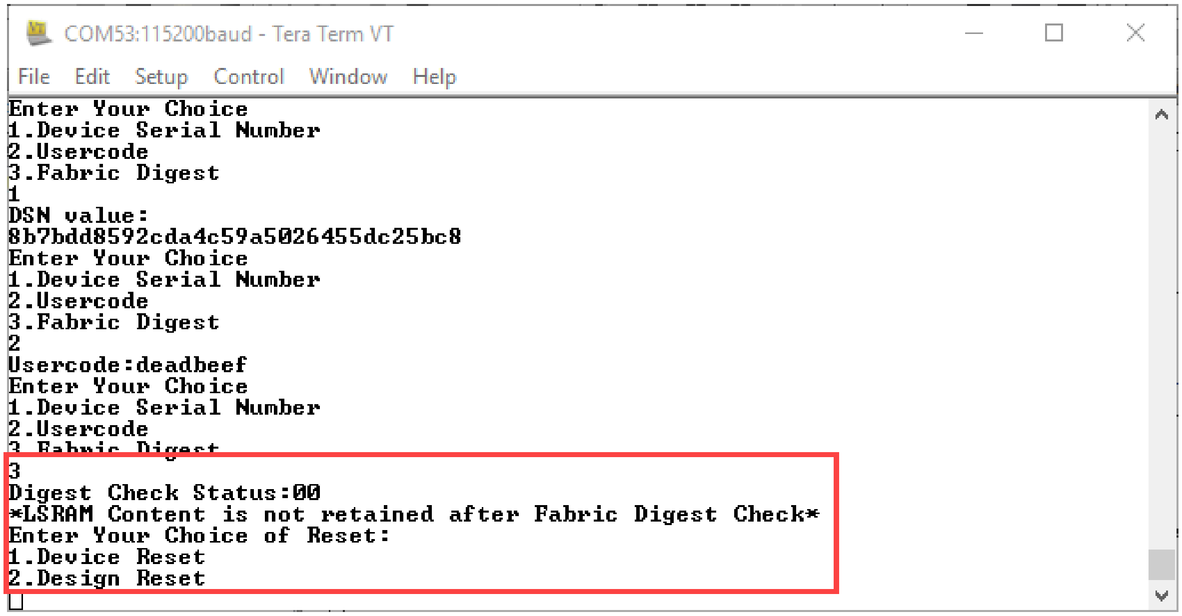
2. Press '2' to execute Fetch JTAG USERCODE system service. As a result, the USERCODE or silicon signature programmed in the device is printed on the terminal, as shown in the following figure.

   Figure 1-12. Fetch JTAG USERCODE System Service



3. Press '3' to perform a digest check on the FPGA fabric. Notice that the user LEDs on the board turns off while the fabric digest check is being performed and then resume counting from the previous state after exiting from suspend state.
   The status of the fabric digest check is returned by the service, and it is printed on the terminal.
   A Status code of 0x00 means no error in the fabric digest check.

Figure 1-13. Fabric Digest Check System Service



4. Press '1' to issue a device reset. Notice that the LEDs start counting from 0x0 since the counter is being reset. Pressing "2" instead of "1" resets the design, but the counter continues counting without resetting. In either case, the rest of the design restarts and the terminal shows the start menu.

This concludes the demo of Fabric Digest Check system service.

2. Appendix: Running the TCL Script

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

Perform the following steps to run a TCL script:

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 the TCL script, the Libero project is created in the TCL_Scripts directory.

For more information about TCL scripts, refer to **mpf_an4593_v2022.1_eval_df/TCL_Scripts/readme.txt**.

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

3. Revision History

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

Revision	Date	Description
A	06/2022	The following is the list of changes in revision A of the document: • The document was migrated to the Microchip template. • The document number was updated from 51900482 to DS00004593. • The document ID was updated to AN4593 from AC482. • Updated the document for Libero SoC v2022.1. • Updated the Figure 1-6. • Updated the Table 1-1. • Updated the note in 1.3. Prerequisites.
4.0	—	Information about CoreSysServices_PF IP core was replaced with PF_SYSTEM_SERVICES IP core.
3.0	—	Added 2. Appendix: Running the TCL Script.
2.0	—	The following is a summary of the changes made in this revision. • Updated the document for Libero SoC v12.2. • Removed the references to Libero version numbers.
1.0	—	Initial release.

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

Trademarks

ISBN: 978-1-6683-0745-8

Quality Management System

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

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 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