Stratix 10 Mailbox Client Intel FPGA IP Core Design Example(QSPI Flash Access and Remote System Update)

Date: March 2021

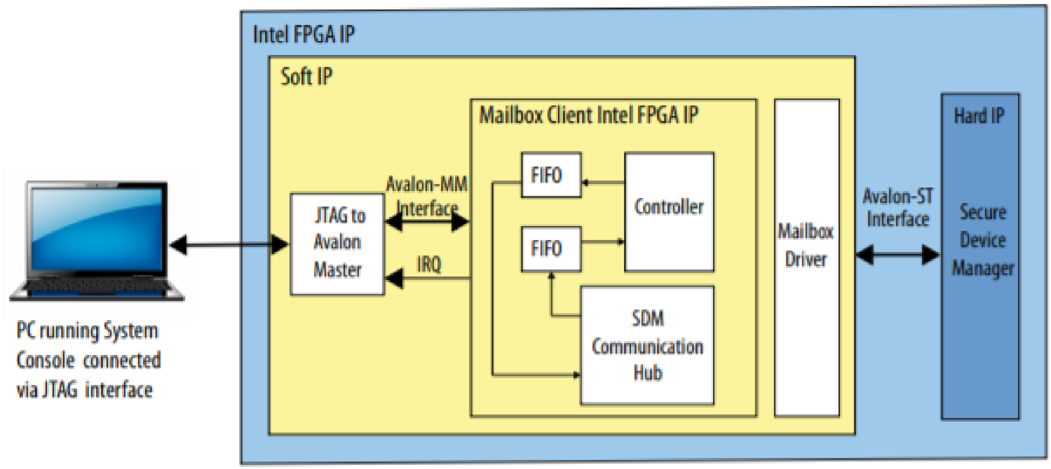
Revision: 2021.3.10

1 Contents

1 Introduction/Overview

This reference design implements the Mailbox Client Intel FPGA IP core in Stratix 10 FPGA.

Figure 1: Mailbox Client Intel FPGA IP System Block Diagram



The Mailbox Client Intel FPGA IP is a bridge between a host and the Secure Device Manager(SDM). You use the Mailbox Client Intel FPGA IP to send commands and receive status from SDM peripheral clients. The Mailbox Client Intel FPGA IP is an Avalon MM slave component that must connect to an Avalon MM master.

In this reference design, JTAG-to-Avalon Master acts as the host controller connecting to Mailbox Client Intel FPGA IP core. The JTAG-to-Avalon Master Bridge IP translates the commands it received from the System Console to Avalon Memory-Mapped (Avalon MM) format that the Mailbox Client Intel FPGA IP requires. Mailbox Client Intel FPGA IP drives commands and receives responses from the SDM.
The rsu1.tcl script provides examples to perform the available commands supported by SDM. You can run the functions available in the rsu1.tcl script via System Console of the Intel Quartus Prime software to perform the following operations,

- Read FPGA IDCODE
- Read FPGA CHIP ID
- QPSI flash access operations such as read/write/erase on QSPI flash and program QSPI flash with .rpd file.
- Remote System Update(RSU) operations such as reading RSU status and trigger reconfiguration from the data source which can be either an application image or factory image.

The rsu1.tcl script can be downloaded from the link provided below.
https://www.intel.com/content/dam/altera-www/global/en_US/others/support/devices/configuration/rsu1.tcl

The following sections present a complete remote system update example, including the following steps. In this example, various QSPI flash access operations are performed when adding new application image in QSPI flash.

1. Creating the initial remote system update image (.jic) containing the bitstreams for the factory image and one application image.
2. Programming the flash memory with the initial remote system update image that subsequently configures the device.
3. Reconfiguring the device with an application or factory image.
4. Creating a single remote system update (.rpd) containing the bitstreams to add an application image in user mode.
5. Adding an application image.
6. Removing an application image.

2 Requirements

2.1 Hardware and Software Requirements

To run this remote system update example, your system must meet the following hardware and software requirements:

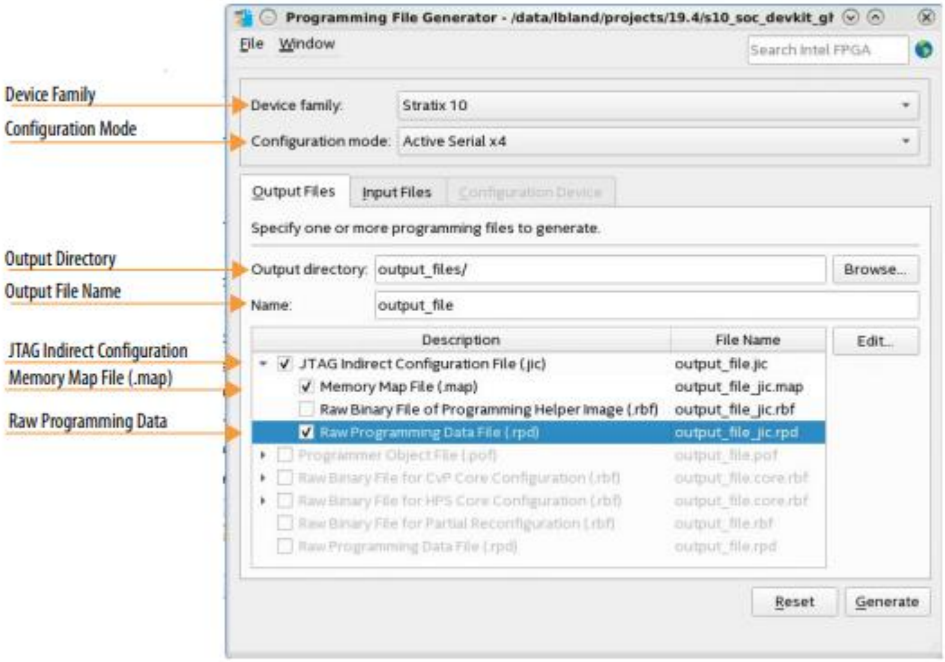
- Intel Quartus Prime Pro Edition software version 20.4 or later
- Stratix 10 Mailbox Client Intel FPGA IP Core Design Example(QSPI Flash Access and Remote System Update) from Intel Design Store
- Stratix 10 GX L-Tile Development Kit

3 Walkthrough

3.1 Creating Initial Flash Image Containing Bitstreams for Factory Image and One Application Image

1. On the File menu, click Programming File Generator.
2. Select Intel Stratix 10 from the Device family drop-down list.
3. Select the configuration mode from the Configuration mode drop-down list. The current Intel Quartus Prime Software only supports remote system update feature in Active Serial x4.
4. On the Output Files tab, assign the output directory and file name.
5. Select the output file type. Select the following file types for the Active Serial (AS) x4 configuration mode:
    a. JTAG Indirect Configuration File (.jic)
    b. Memory Map File (.map)
    c. Raw Programming File (.rpd). Generating the .rpd file is optional.

Figure 2: Creating Initial Flash Image



Device family: Stratix 10

Configuration mode: Active Serial x4

Specify one or more programming files to generate.

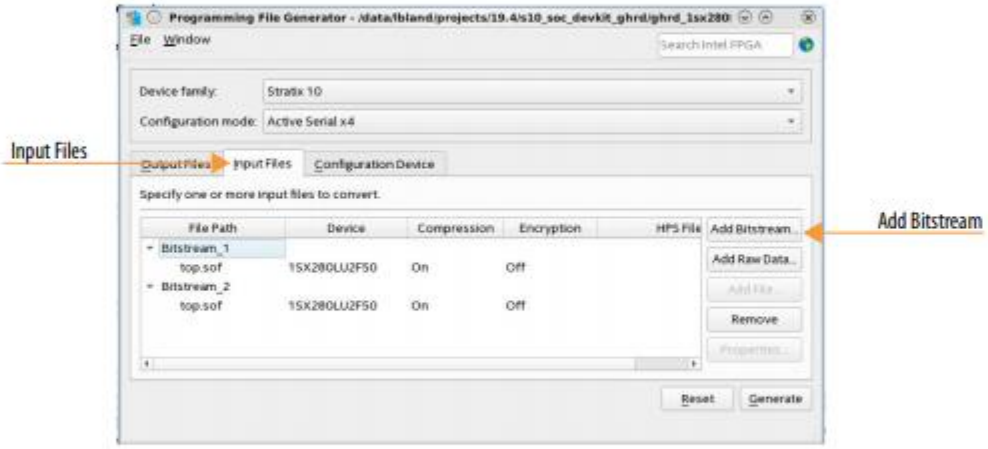
Output directory: output_files/

Name: output_file

Description	File Name
☑ JTAG Indirect Configuration File (.jic)	output_file.jic
☑ Memory Map File (.map)	output_file_jic.map
□ Raw Binary File of Programming Helper Image (.rbf)	output_file_jic.rbf
☑ Raw Programming Data File (.rpd)	output_file_jic.rpd
□ Programmer Object File (.pof)	output_file.pof
□ Raw Binary File for CvP Core Configuration (.rbf)	output_file.core.rbf
□ Raw Binary File for HPS Core Configuration (.rbf)	output_file.core.rbf
□ Raw Binary File for Partial Reconfiguration (.rbf)	output_file.rbf
□ Raw Programming Data File (.rpd)	output_file.rpd

6. On the Input Files tab, click Add Bitstream, select the factory and application image .sof files and click Open.
    a. Bitstream_1 is the bitstream for factory image.
    b. Bitstream_2 is the bitstream for application image.

Figure 3: Input Files Tab: Specifying the .sof



Device family: Stratix 10

Configuration mode: Active Serial x4

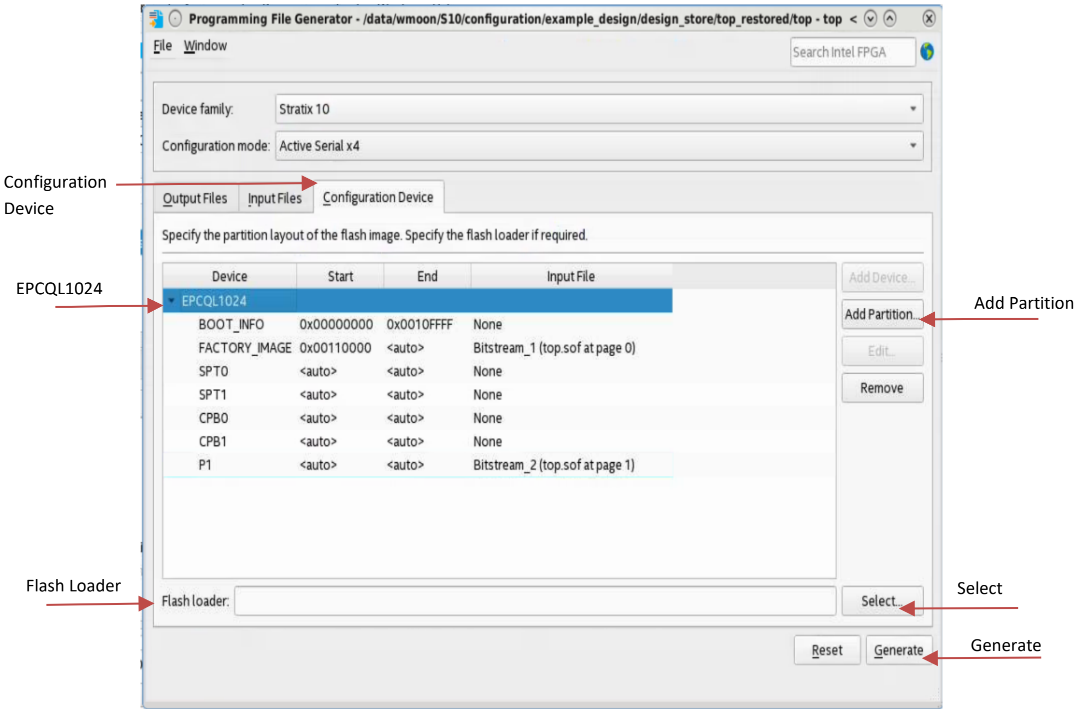
Specify one or more input files to convert.

File Path	Device	Compression	Encryption	HPS File
Bitstream_1				
top.sof	1SX280LU2F50	On	Off	
Bitstream_2				
top.sof	1SX280LU2F50	On	Off	

7. On the Configuration Device tab, click Add Device, select EPCQL1024 flash memory and click OK. The Programming File Generator tool automatically populates the flash partitions.
8. Select the FACTORY_IMAGE partition and click Edit.

9. On the Edit Partition dialog box, select Bitstream_1 as the factory image .sof in the Input file drop-down list. Keep the default settings for the Page and Address Mode. Click OK.
10. Select the EPCQL1024 flash memory and click Add Partition.
11. In the Add Partition dialog box, select Bitstream_2 for the application image .sof in the Input file drop-down list. Assign Page: 1.Keep the default settings for Address Mode. Click OK.
12. For Flash loader click Select. Select Intel Stratix 10 from Device family list. Select 1SG280LU2 for the Device name. Click OK.
13. Click Generate to generate the remote system update programming files. The Programming File Generator generates the following files:
    a. Initial_RSU_Image.jic
    b. Initial_RSU_Image_jic.map

Figure 4: Configuration Tab: Add Device, Partition, Flash Loader and Generate



Device family: Stratix 10

Configuration mode: Active Serial x4

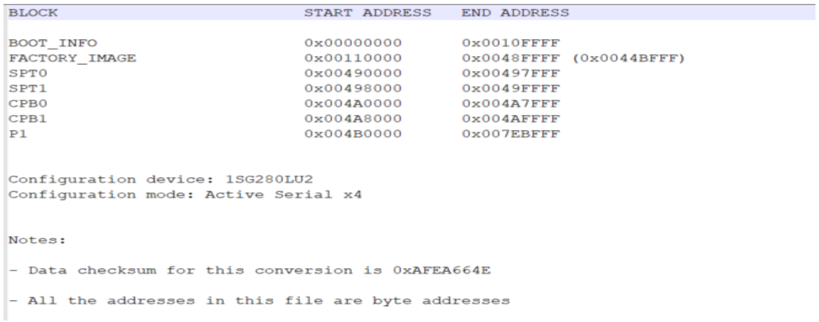
Specify the partition layout of the flash image. Specify the flash loader if required.

Device	Start	End	Input File
EPCQL1024			
BOOT_INFO	0x00000000	0x0010FFFF	None
FACTORY_IMAGE	0x00110000	<auto>	Bitstream_1 (top.sof at page 0)
SPT0	<auto>	<auto>	None
SPT1	<auto>	<auto>	None
CPB0	<auto>	<auto>	None
CPB1	<auto>	<auto>	None
P1	<auto>	<auto>	Bitstream_2 (top.sof at page 1)

Flash loader:

The following example output shows the generated .map file. The .map lists the start addresses of the factory image, CPB0, CPB1, and one application image. The remote system update requires these addresses.

Figure 5: Example of the contents in the generated .map file



```
BLOCK                                START ADDRESS     END ADDRESS

BOOT_INFO                            0x00000000        0x0010FFFF
FACTORY_IMAGE                        0x00110000        0x0048FFFF (0x0044BFFF)
SPT0                                 0x00490000        0x00497FFF
SPT1                                 0x00498000        0x0049FFFF
CPB0                                 0x004A0000        0x004A7FFF
CPB1                                 0x004A8000        0x004AFFFF
P1                                   0x004B0000        0x007EBFFF


Configuration device: 1SG280LU2
Configuration mode: Active Serial x4


Notes:

- Data checksum for this conversion is 0xAFEA664E

- All the addresses in this file are byte addresses
```

After generating the programming file, you can program the flash memory.

3.2 Programming Flash Memory with the Initial Remote System Update Image

You can program the initial remote system update image from the command line. In the following command, substitute your .jic for output_file.jic if necessary.

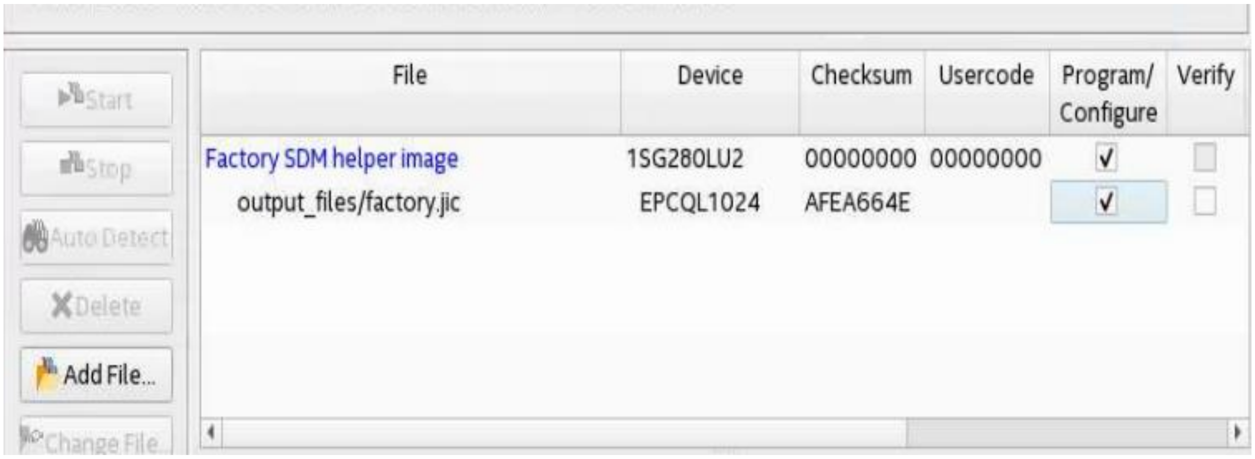
```
quartus_pgm -c 1 -m jtag -o pvbi\;./output_file.jic
```

Critical note: You must use INI option “**PGM_QSPI_FAST_PROGRAM=OFF”** if the on board configuration device is an EPCQL device.

Alternatively, you can use the Intel Quartus Prime Programmer to program the initial RSU update image by completing the following procedure:

1. Open the Programmer and click Add File. Select the generated .jic file (output_file.jic) and click Open.
2. Turn on the Program/Configure for the attached .jic file.
3. To begin programming the flash memory with the initial remote system update image, click Start.
4. Configuration is complete when the progress bar reaches 100%. Power cycle the board to automatically configure the Intel Stratix 10 device with the application image using the AS x4 configuration scheme.

Figure 6: Programming the Flash Memory with the Initial RSU Image



File	Device	Checksum	Usercode	Program/ Configure	Verify
Factory SDM helper image	1SG280LU2	00000000	00000000	☑	□
output_files/factory.jic	EPCQL1024	AFEA664E		☑	□

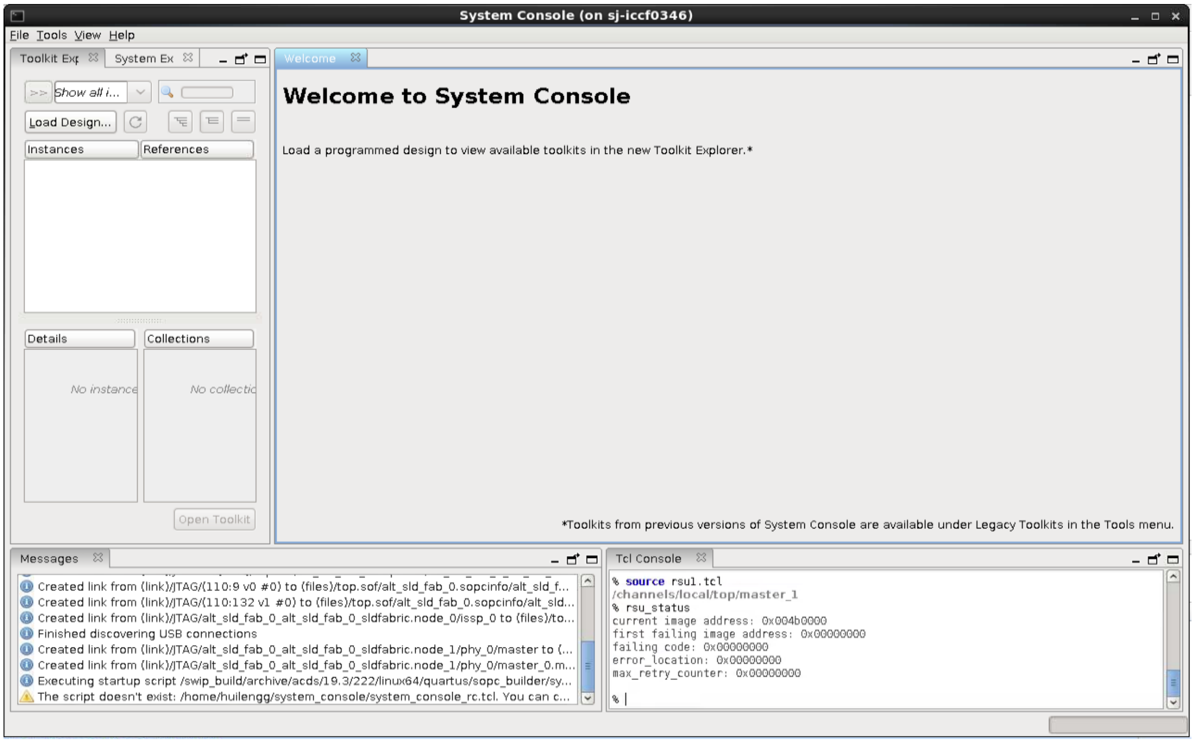
Note: This example does not assign the Direct to Factory Image pin. Consequently, the Programmer configures the device with the application image. The application image is the default image if the design does not use the Direct to Factory Image pin.

5. Use the RSU_STATUS command to determine which bitstream image the Programmer is using as shown in the following example:
    a. In the Intel Quartus Prime software, select Tools ➤ System Debugging Tools ➤ System Console to launch the system console.
    b. In the Tcl Console pane, type source rsu1.tcl to open the example of Tcl script to perform the remote system update commands. Refer to the Related Information for a link to rsu1.tcl.
        i. Note: Make sure you assign the correct variable in the list returned by get_service_paths. For example, you need to assign mp to 0 as shown in below to get the proper service path.

```
set mp [lindex [get_service_paths master] 0]
```

    c. Type the rsu_status command to report the current remote system update status. You can retrieve the current running image address from the remote system update status report. The current image address must match the start address for the application image printed in the .map file.

Figure 7: Running Tcl Commands Available in rsu1.tcl



Welcome to System Console

Load a programmed design to view available toolkits in the new Toolkit Explorer.*

*Toolkits from previous versions of System Console are available under Legacy Toolkits in the Tools menu.

```
Created link from (link)/JTAG/(110:9 v0 #0) to (files)/top.sof/alt_sld_fab_0.sopcinfo/alt_sld_f...
Created link from (link)/JTAG/(110:132 v1 #0) to (files)/top.sof/alt_sld_fab_0.sopcinfo/alt_sld...
Created link from (link)/JTAG/alt_sld_fab_0_alt_sld_fab_0_sldfabric.node_0/issp_0 to (files)/to...
Finished discovering USB connections
Created link from (link)/JTAG/alt_sld_fab_0_alt_sld_fab_0_sldfabric.node_1/phy_0/master to (...
Created link from (link)/JTAG/alt_sld_fab_0_alt_sld_fab_0_sldfabric.node_1/phy_0/master_0.m...
Executing startup script /swip_build/archive/acds/19.3/222/linux64/quartus/sopc_builder/sy...
The script doesn't exist: /home/huilengg/system_console/system_console_rc.tcl. You can c...
```

```
% source rsu1.tcl
/channels/local/top/master_1
% rsu_status
current image address: 0x004b0000
first failing image address: 0x00000000
failing code: 0x00000000
error_location: 0x00000000
max_retry_counter: 0x00000000

% |
```

3.3 Reconfiguring the Device with an Application or Factory Image

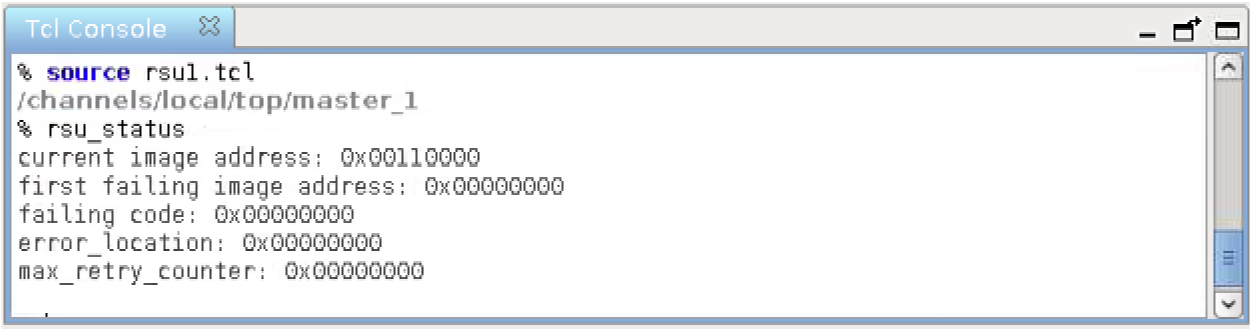
The following steps describe the process to reconfigure the device with a different application image or the factory image using operation commands after the device is in user mode.

1. The remote system update host sends the RSU_IMAGE_UPDATE command to perform the remote system update to the new application image or factory image.
    a. For example, in the Tcl console of the System Console, type the following command to initiate a remote system update to the factory image.
        i. rsu_image_update 0x00110000
        This command reconfigures the device with factory image. Address 0x00110000 is the start address of the factory image as shown in the .map file. The JTAG host automatically disconnects from the System Console once the device reconfiguration is successful. You must restart the System Console to re-establish the connection with the device to perform next command.
        ii. rsu_image_update 0x004B0000
        This command reconfigures the device with the application image. Address 0x004B0000 is the start address of the application image as shown in the .map file.

    Optional: Retrieve the remote system update status by using the rsu_status command to ensure you have successfully reconfigured the device.

2. In the Tcl console of the System Console, type rsu_status to verify the current image. The following figure shows the device is being reconfigured with the factory image.

Figure 8: Verify Current Image Using rsu_status Command



```
% source rsu1.tcl
/channels/local/top/master_1
% rsu_status
current image address: 0x00110000
first failing image address: 0x00000000
failing code: 0x00000000
error_location: 0x00000000
max_retry_counter: 0x00000000
```

3.4 Adding an Application Image

Complete the following steps to add an application image to flash memory:

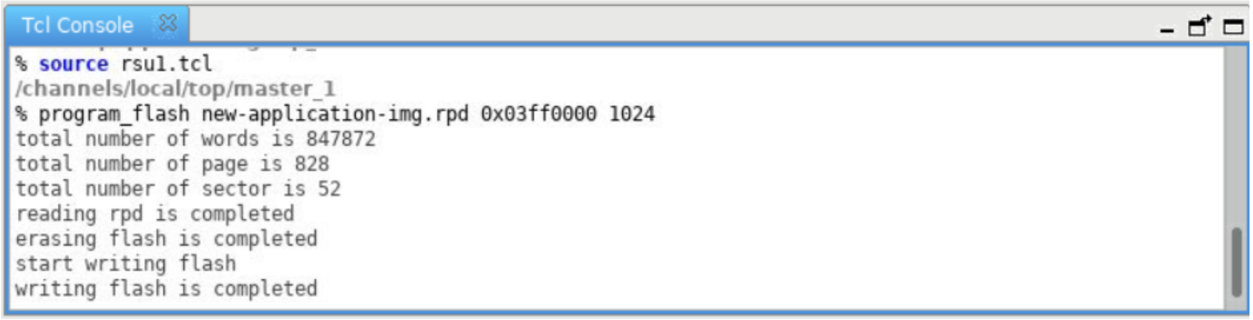
1. Set up exclusive access to the AS x4 interface and flash memory by running the QSPI_OPEN and QSPI_SET_CS commands in the Tcl Console window. You now have exclusive access to the AS x4 interface and flash until you relinquish access by running the QSPI_CLOSE command. Write the new application image to the flash memory using the QSPI_WRITE command.
2. Alternatively, the rsu1.tcl script includes the program_flash function that programs a new application image(.rpd file) into flash memory. The following command accomplishes this task:

```
program_flash new_application_image.rpd 0x03FF0000 1024
```

The program_flash function takes three arguments:

   a. The .rpd file to write to flash memory.
   b. The start address.
   c. Number of words to write for each QSPI_WRITE command. The QSPI_WRITE supports up to 1024 words per write instruction.

Figure 9: Program New Application Image



```
% source rsu1.tcl
/channels/local/top/master_1
% program_flash new-application-img.rpd 0x03ff0000 1024
total number of words is 847872
total number of page is 828
total number of sector is 52
reading rpd is completed
erasing flash is completed
start writing flash
writing flash is completed
```

3. Write the new application image start address to a new image pointer entry in the configuration firmware pointer block (CPB) using the QSPI_WRITE command. Ensure that the new image pointer entry value is 0xFFFFFFFF before initiating the write.

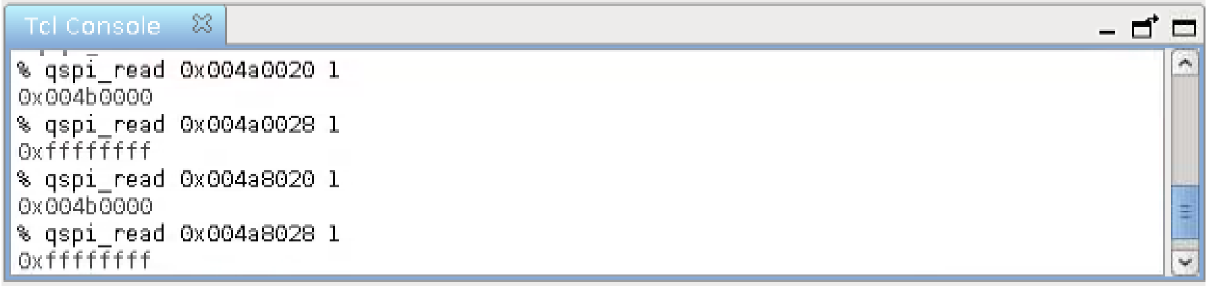
Note: You must update both copies (CBP0 and CBP1) when editing the configuration firmware pointer block. Refer to Pointer Block Layout table in Intel® Stratix 10® Configuration User Guide for more details.

Based on the example described above, the address offset 0x20 in the CPB0 and CPB1 must point to the start address of the application image. The next new image pointer entry value must be 0xFFFFFFFF before you write the start address of the new application image to the next image pointer entry.

You can use the QSPI_read function to verify that the new image pointer entry value is 0xFFFFFFFF . The QSPI_read function takes in two arguments:

1. Start address
2. Number of words to read

Figure 10: Verifying that the New Image Pointer entry Value is 0xFFFFFFFF

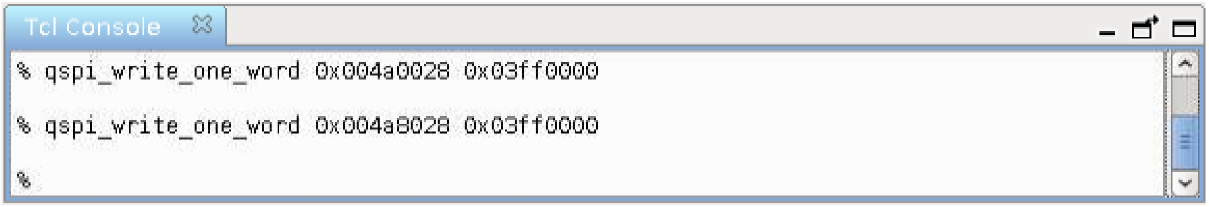


```
% qspi_read 0x004a0020 1
0x004b0000
% qspi_read 0x004a0028 1
0xffffffff
% qspi_read 0x004a8020 1
0x004b0000
% qspi_read 0x004a8028 1
0xffffffff
```

You can now proceed to write the new application image address to next image entry by using the QSPI_write_one_word function. The QSPI_write_one_word function takes in two arguments:

1. Address
2. The value of the word

Figure 11: Writing an Address Pointer to the New Image Pointer Entry



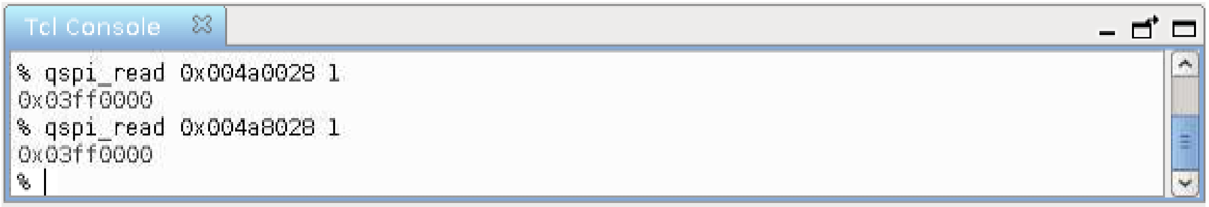
```
% qspi_write_one_word 0x004a0028 0x03ff0000

% qspi_write_one_word 0x004a8028 0x03ff0000

%
```

You can now do a QSPI_read function to the next image pointer entry to ensure that it is written with the start address of the desired new application image.

Figure 12: Verifying the Update to the New Image Pointer



```
% qspi_read 0x004a0028 1
0x03ff0000
% qspi_read 0x004a8028 1
0x03ff0000
% |
```

Host software can now reconfigure the Intel Stratix 10 FPGA with the new application image by asserting the nCONFIG pin. Alternatively, you can power cycle the PCB. After reconfiguration, check the current image address. The expected address is 0x03ff0000. After adding a new image, your application image list includes the newly added application image and the old application image, which is now a secondary image. The newly added application image has the highest priority.

Note: When the remote system update host loads an application image, the decision firmware traverses the image pointer entries in reverse order. The new image has the highest priority when you restart the device.

3.5 Removing an Application Image

1 Set up exclusive access to the AS x4 interface and flash memory by running the QSPI_OPEN and QSPI_SET_CS commands in the Tcl Console window. You now have exclusive access to the AS x4 interface and flash until you relinquish access by running the QSPI_CLOSE command. Write the new application image to the flash memory using the QSPI_WRITE command.

2 Write 0x00000000 to the application image start address stored in the image pointer entry of the configuration firmware pointer block (CPB0 and CPB1) using the QSPI_WRITE command. Note: You must update both copies (copy0 and copy1) when editing the configuration firmware pointer block.

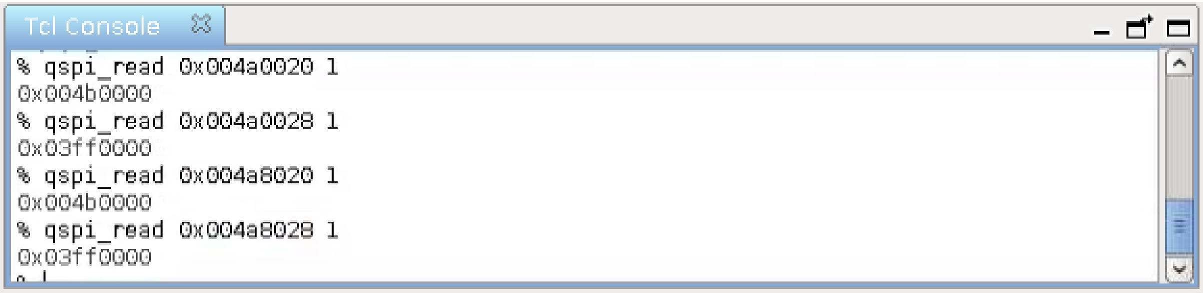
3 Erase the application image content in the flash memory using the QSPI_ERASE command.

4 To remove a new application image, add another new application image in the next or subsequent image pointer entry or allow the device to fall back to the previous or secondary application image in your application image list. The following table shows correct entries for image pointer entries for CPB0 and CPB1 for offsets 0x20 and 0x28.

Table1: Configuration Firmware Pointer Block Contents

CPB Start Address + 0x20	Content	Value
CPB0 + 0x20 = 0x004A0020	Old application image pointer entry (lower priority)	0x004B0000
CPB0 + 0x28 = 0x004A0028	Current/new application image pointer entry (highest priority)	0x03FF0000
CPB1 + 0x20 = 0x004A8020	Old application image pointer entry (lower priority)	0x004B0000
CPB1 + 0x28 = 0x004A8028	Current/new application image pointer entry (highest priority)	0x03FF0000

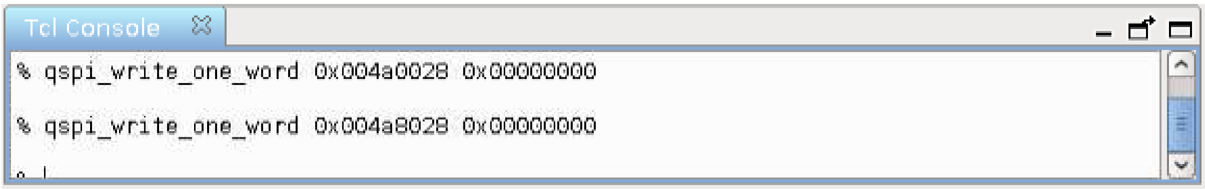
Figure 13: Read Current CPB Values



```
% qspi_read 0x004a0020 1
0x004b0000
% qspi_read 0x004a0028 1
0x03ff0000
% qspi_read 0x004a8020 1
0x004b0000
% qspi_read 0x004a8028 1
0x03ff0000
```

You can now remove the current or new application image address image pointer entry by writing the value to 0x00000000 using the QSPI_write_one_word function as shown in the following example. The QSPI_write_one_word function takes address and data arguments. Be sure to erase the application content that you just removed from flash memory.

Figure 14: Remove Application Image

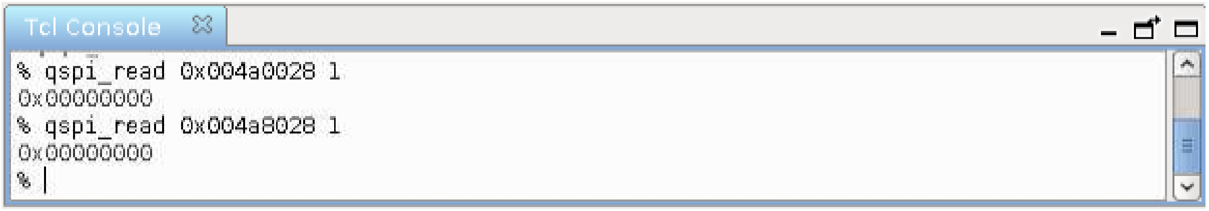


```
% qspi_write_one_word 0x004a0028 0x00000000

% qspi_write_one_word 0x004a8028 0x00000000
```

You can use a QSPI_read to the image pointer entry at offset 0x28 for CBP0 and CPB1 to verify completion of the QSPI_write_one_word commands.

Figure 15: Verify the Writes



```
% qspi_read 0x004a0028 1
0x00000000
% qspi_read 0x004a8028 1
0x00000000
% |
```

You can now configure the device with the old application image. The old application image has the highest priority if you power cycle the device or the host asserts the nCONFIG pin. You can run the rsu_status report to check the status of the current image address.

3.6 References

1 Refer to Mailbox Client Intel FPGA IP User Guide
https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-20087.pdf

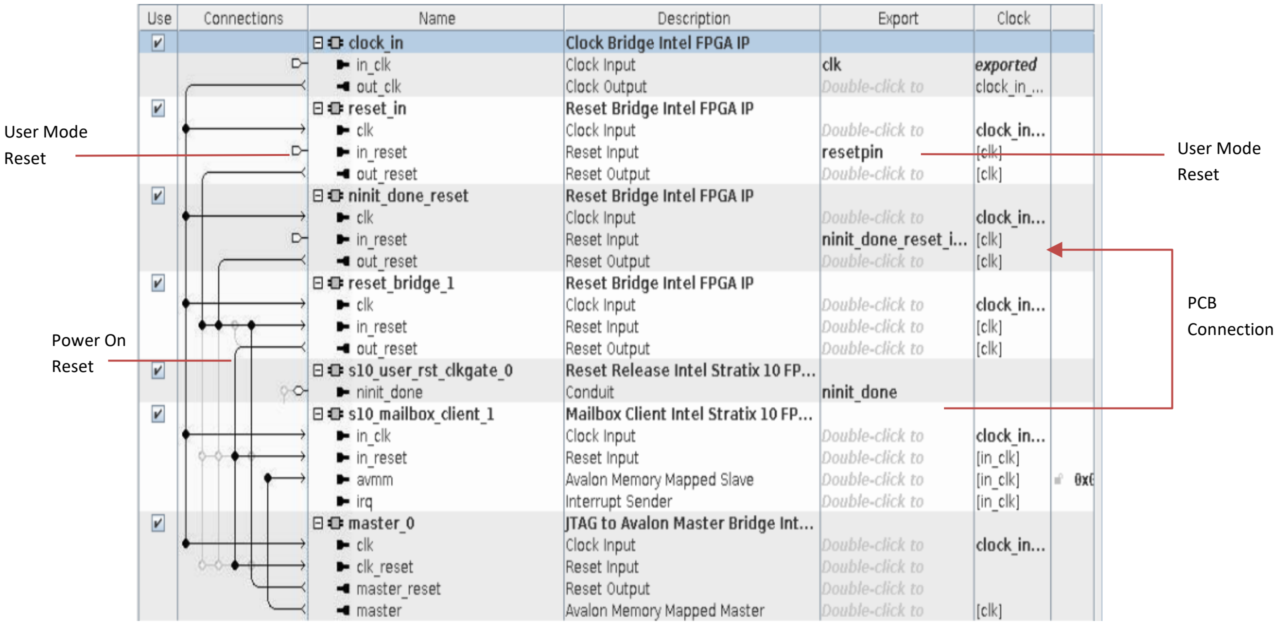
2 Refer to Chapter 5. Remote System Update(RSU) in Intel Stratix 10 Configuration User Guide
https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/stratix-10/ug-s10-config.pdf

4 Concept

4.1 Prerequisites for Design in Quartus

- Your design should include the Mailbox Client Intel FPGA IP that connects to a JTAG to Avalon Master Bridge as shown in the Platform Designer system.
- The JTAG to Avalon Master Bridge acts as the remote system update host controller for your factory and application images.
- In addition, your design must include the Reset Release Intel FPGA IP. This component holds the design in reset until the entire FPGA fabric has entered user mode.
- The ninit_done_reset and reset_bridge_1 components create a two-stage reset synchronizer to release the Mailbox Client Intel FPGA IP and JTAG to Avalon Master Bridge Intel FPGA IP from reset when the device configuration is complete and the device is in user mode.
- The ninit_done output signal from Reset Release IP gates this reset by connecting to the ninit_done_reset_in_reset_reset pin.
- The reset_in Reset Bridge Intel FPGA IP provides a user mode reset. In this design, the exported resetpin connects to application logic.

Figure 16: Required Communication and Host Components for the Remote System Update Design Example



Use	Connections	Name	Description	Export	Clock	
☑		⊟ clock_in	**Clock Bridge Intel FPGA IP**			
		in_clk	Clock Input	clk	*exported*	
		out_clk	Clock Output	*Double-click to*	clock_in_...	
☑		⊟ reset_in	**Reset Bridge Intel FPGA IP**			
		clk	Clock Input	*Double-click to*	clock_in...	
		in_reset	Reset Input	resetpin	[clk]	
		out_reset	Reset Output	*Double-click to*	[clk]	
☑		⊟ ninit_done_reset	**Reset Bridge Intel FPGA IP**			
		clk	Clock Input	*Double-click to*	clock_in...	
		in_reset	Reset Input	ninit_done_reset_i...	[clk]	
		out_reset	Reset Output	*Double-click to*	[clk]	
☑		⊟ reset_bridge_1	**Reset Bridge Intel FPGA IP**			
		clk	Clock Input	*Double-click to*	clock_in...	
		in_reset	Reset Input	*Double-click to*	[clk]	
		out_reset	Reset Output	*Double-click to*	[clk]	
☑		⊟ s10_user_rst_clkgate_0	**Reset Release Intel Stratix 10 FP...**			
		ninit_done	Conduit	ninit_done		
☑		⊟ s10_mailbox_client_1	**Mailbox Client Intel Stratix 10 FP...**			
		in_clk	Clock Input	*Double-click to*	clock_in...	
		in_reset	Reset Input	*Double-click to*	[in_clk]	
		avmm	Avalon Memory Mapped Slave	*Double-click to*	[in_clk]	0x
		irq	Interrupt Sender	*Double-click to*	[in_clk]	
☑		⊟ master_0	**JTAG to Avalon Master Bridge Int...**			
		clk	Clock Input	*Double-click to*	clock_in...	
		clk_reset	Reset Input	*Double-click to*		
		master_reset	Reset Output	*Double-click to*		
		master	Avalon Memory Mapped Master	*Double-click to*	[clk]	

5 Document Revision History

List the revision history for the application note.

Date	Version	Changes
March 2021	2021.3.10	Initial release