AN 010: Using the Internal Reconfiguration Feature to Remotely Update Trion and Titanium FPGAs

AN010-v2.4
June 2023
www.efinixinc.com

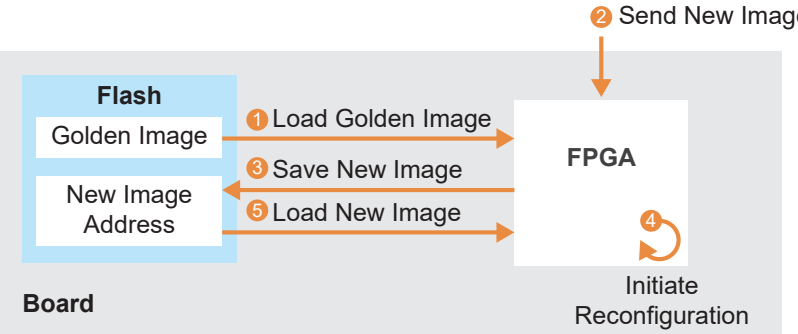
Contents

Introduction

Trion® and Titanium FPGAs have built-in hardware that supports an internal reconfiguration feature in which the FPGA can reconfigure itself from a bitstream image stored in flash memory. This feature is useful for performing system upgrades from a remote location. In these applications, the FPGA is the "brain" that controls the system functionality (that is, there is no microcontroller or CPU).

In a typical reconfiguration flow, the FPGA first configures itself using a "golden" bitstream image as usual. Then, the FPGA receives a new application image remotely, via Ethernet, Wi-Fi, etc., and saves it to flash memory. Then, the FPGA triggers itself to reconfigure using the new image.

Figure 1: Performing Remote Update Using Internal Reconfiguration



The following FPGAs support the internal reconfiguration feature.

Table 1: Internal Reconfiguration Support

FPGA	Package	Supported
T4	BGA49, BGA81	-
T8	BGA49, BGA81	-
	QFP144	✓
T13, T20, T35, T55, T85, T120	All	✓
Ti35, Ti60, Ti90, Ti120, Ti180	All	✓

Planning for Remote Update

For remote update to work, your user design should be able to receive a new application image from an outside source, send it to the FPGA, and write it to a location in the flash memory. When planning your system for remote updating, consider these guidelines:

- Choose your flash device size to accommodate future updates.
- Ensure that the address ranges you choose for the images can accommodate future updates. (Refer to **Writing to Flash Memory** on page 9 for more details.)
- Implement a controller to obtain the new application image from an outside source.
- Incorporate a SPI flash master block into your design to write and verify application images in flash memory.
- Ensure that any dual-purpose configuration pins used as output adhere to the signal transition time requirement. (Refer to **Dual-Purpose Configuration Pin Requirement** on page 4 for more details.)

Dual-Purpose Configuration Pin Requirement

Some configuration pins are dual-purpose. These pins can be configured as a general-purpose input/output (GPIO) during user mode but must be set to the appropriate condition during reconfiguration. As GPIO output, these signals can be used to drive external networks such as passive components or external devices.

Table 2: Dual-Purpose Configuration Pins by Family

Pin	Trion	Titanium
CBUS2, CBUS1, CBUS0	✓	-
SS_N	✓	-
SSL_N	-	✓
TEST_N	✓	✓

Important: You must ensure that this requirement is met when using any of the dual-purpose configuration pins as output during user mode.

When the internal reconfiguration is triggered, all GPIOs including the dual-purpose configuration pins that are configured as output are driven high or low depending on the design. These dual-purpose configuration pins must be then set to the appropriate condition before reconfiguring with a new image. The high-to-low or low-to-high transition time can be impacted by the external network connecting to the dual-configuration pins. The rising-time and falling-time of the external network must fall within 800 ns to ensure that the configuration pins are stable and the reconfiguration is successful.

Figure 2: Trion Dual-Purpose Configuration Pins to External Network Connections

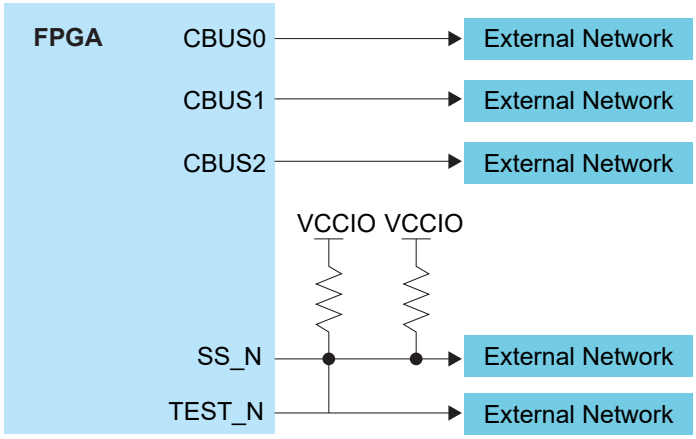
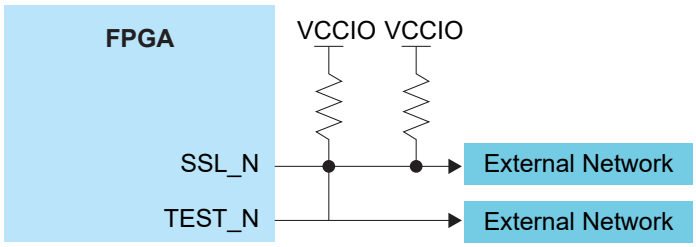


Figure 3: Titanium Dual-Purpose Configuration Pins to External Network Connections



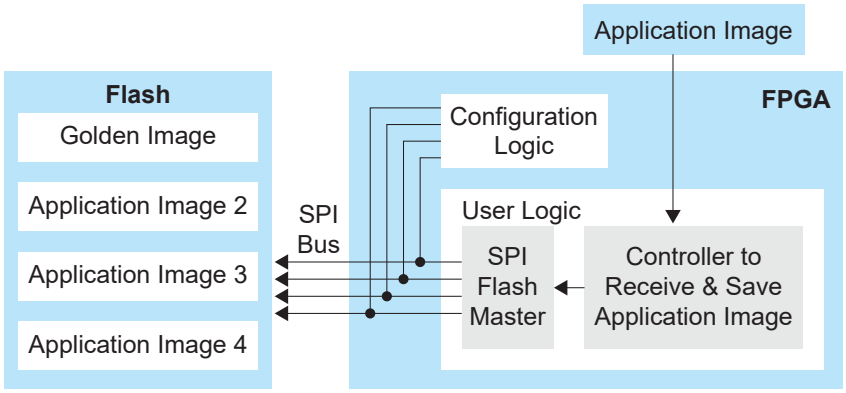
Building the Golden Design

In addition to user logic, the golden design should:

- Enable the reconfiguration interface in the Efinity® Interface Designer.
- Include a trigger to initiate reconfiguration.
- Use active configuration mode.
- Receive and save a new application image to the flash memory.

Note: Refer to AN 006: Configuring Trion FPGAs or AN 033: Configuring Titanium FPGAs for more information on using active configuration mode.

Figure 4: Receiving New Image and Saving to Flash



The user logic and configuration logic utilize the same SPI bus to access the images in the flash memory.

Enable the Internal Reconfiguration Interface

You enable the interface in the Interface Designer.

To enable internal reconfiguration:

1. Click **Device Setting > Configuration**.
2. In the Block Editor, turn on **Enable Internal Reconfiguration Interface**.
3. In the Block Editor **Remote Update** tab, turn on **Enable Internal Reconfiguration Interface**.
4. Indicate the name of the clock pin that will control the internal reconfiguration.
5. Define the FPGA pins that the interface uses.
6. Save.

Internal Reconfiguration Signals

The internal reconfiguration interface uses four control signals and a signal that selects the location of the new image. When you enable the internal reconfiguration interface in the Interace Designer, the software adds these signals to your design.

Note: The FPGA does not require input via the package pins to trigger reconfiguration.

Figure 5: Internal Reconfiguration Block Diagram

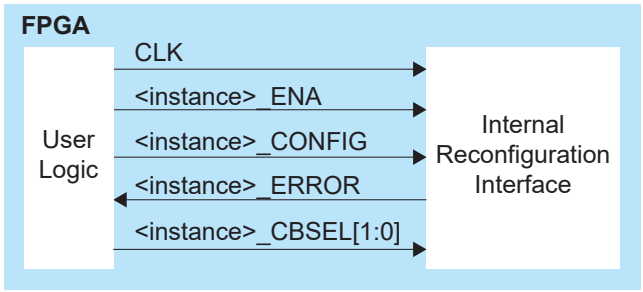


Table 3: Internal Reconfiguration Pins

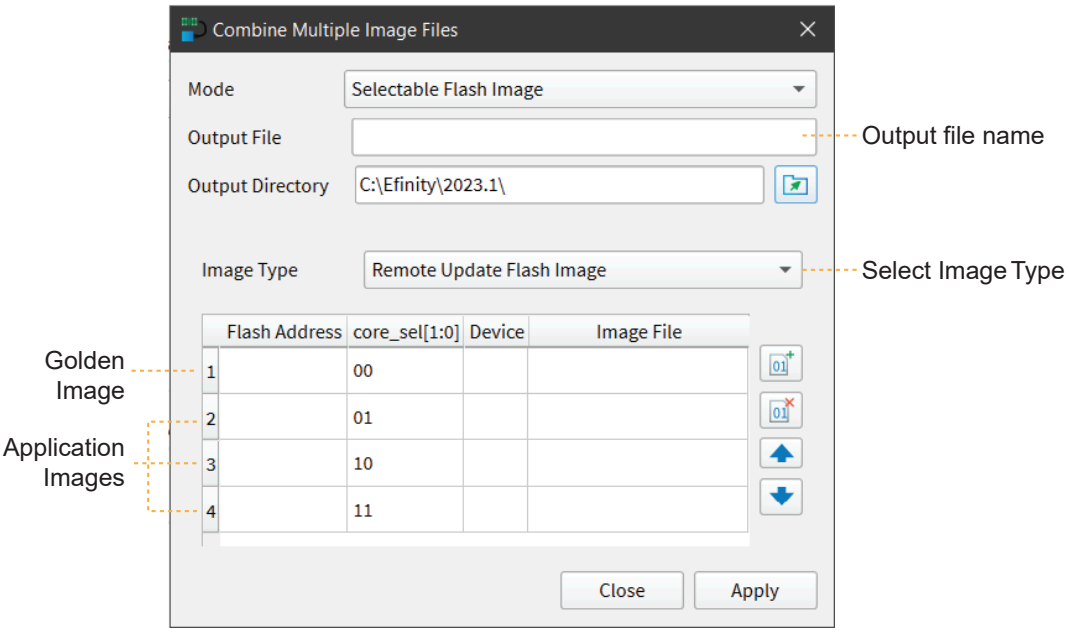
Signal	Direction	Description
<*instance*>_CBSEL[1:0]	Input	Multi-image select signals to the internal reconfiguration interface (not package pins). Use these signals to choose which image to load from flash memory.
CLK	Input	Clock used to latch <*instance*>_CBSEL when ENA is high. The maximum supported frequency is 100 MHz.
<*instance*>_CONFIG	Input	Asynchronous control that initiates reconfiguration.
<*instance*>_ENA	Input	When <instance>_ENA is high, read the value of <instance>_CBSEL.
<*instance*>_ERROR	Output	Status signal. Signal is set to 0 during power-up. 0 if reconfiguration is successful or no reconfiguration triggered. 1 if the reconfiguration failed.(1)

(1) If reconfiguration fails 6 times, the FPGA configures itself using the golden image.

Combining Images

You can store up to 4 images in flash memory and use them to configure Trion or Titanium FPGAs. You use the Efinity® Programmer to combine the images into one bitstream image that you load into the flash device.

Figure 6: Combine Multiple Image Files



1. Open the Combine Multiple Image Files tool in the Efinity® Programmer.
2. Enter the desired output file name and output directory.
3. Select **Remote Update Flash Image** in the image type drop-down list.
4. Select the first row for image file, click on **Add Image**, and add the golden image.
5. Select the subsequent image file row, and add the application image.
6. Click **Apply** to generate the combined image.

If you do not assign flash addresses in the Programmer, the software automatically chooses them using 4096 boundaries. The Programmer reports the assigned addresses in the console and the report file.

Note: To update the flash memory with a new golden image, create the new image and then use the Combine Multiple Image Files tool in the Efinity® Programmer to regenerate a combined bitstream image. You cannot use remote update to change the golden image.

Writing to Flash Memory

After you set up your system to receive an image remotely and communicate with the flash device using the SPI interface, you are ready to process new images. Trion and Titanium FPGAs can read/write to up to 4 addresses in the flash. When using the internal reconfiguration feature, the golden image is assumed to be at address 0. You can use the other 3 addresses for application images.

Note: Even when you reconfigure with a new application image, Efinix recommends that you leave the golden image in the flash at address 0 as a fallback. Otherwise, you run the risk of not being able to communicate remotely with the FPGA in the system.

For many SPI flash devices, the manufacturer recommends that you erase the flash memory before performing writes. You can erase the entire device, or just a sector. Efinix recommends that the starting address for any image is an exact multiple of 4096. This boundary guarantees that you can erase/write any given image without corrupting other images or other data on the flash device.

Flow for Writing a New Application Image

To ensure that the new image is written to the flash correctly, use the following flow:

1. Erase the flash address you want to use.
2. Write the new image.
3. Read back the image and verify that it is correct.

Triggering Reconfiguration

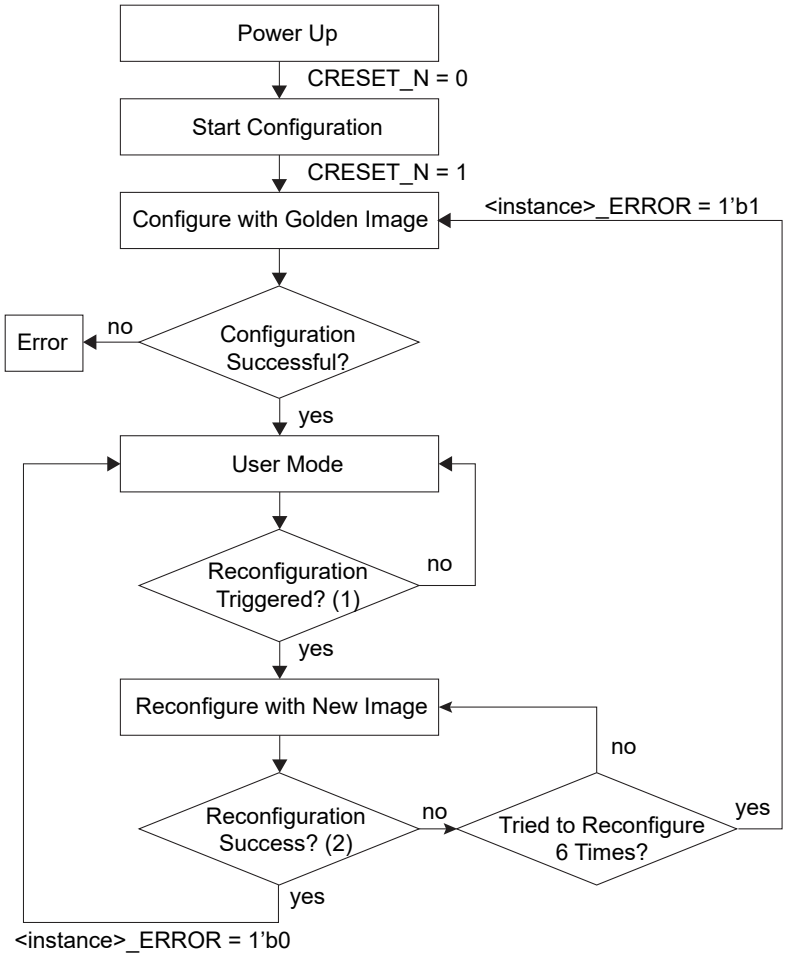
Now that you have the golden image and one or more application images programmed into flash, you can trigger reconfiguration. Upon device power up, the FPGA configures with the golden image and the `CDONE` pin goes high to indicate that configuration is complete. To trigger reconfiguration:

1. Select the image using the <*instance*>`_CBSEL` signals.
2. Hold <*instance*>`_ENA` high.
3. Assert <*instance*>`_CONFIG` high to initiate reconfiguration. The FPGA configures itself using the selected image.
4. Check the value of the <*instance*>`_ERROR` signal. A 0 means successful reconfiguration; a 1 means reconfiguration failed.

Important: The FPGA attempts to reconfigure itself with the new image 6 times. If configuration fails all 6 times, the FPGA falls back to the golden image. The total duration for 6 reconfiguration attempts are up to 41 seconds. Efinix recommends that you monitor the <*instance*>`_ERROR` pin; terminate the reconfiguration process if reconfiguration failed so that the FPGA does not continue the loop of reconfiguring with the golden image.

The configuration fails when the Trion® FPGA detected a CRC error in the configuration RAM (CRAM) bit of the new image received from the flash memory.

Figure 7: Reconfiguration Flow Chart



Note:
1. Trigger reconfiguration with:
   <instance>_CBSEL[1:0] = 2'bXX where XX is the selection of the new image
   <instance>_ENA = 1'b1
   <instance>_CONFIG = 1'b1
2. The FPGA performs a CRC check to ensure that the received image is correct.

Next Steps

Efinix provides an example design that demonstrates how the T20 FPGA can configure itself on the fly with a new image stored in serial NOR flash memory. This design targets the Trion T20 BGA256 development board.

Download: View the example design in the Support Center.

Revision History

Table 4: Revision History

Date	Version	Description
June 2023	2.4	Updated Internal Flash Image GUI name to Remote Update Flash Image. (DOC-1337)
December 2022	2.3	Added maximum supported CLK frequency. (DOC-1028) Added support for Ti90, Ti120, and Ti180 FPGAs.
June 2022	2.2	Updated <instance>_ERROR description. (DOC-817)
February 2022	2.1	Added details on what happens if reconfiguration fails. (DOC-685)
August 2021	2.0	Added support for Titanium FPGAs.
August 2020	1.3	Added the total duration for the 6 reconfiguration attempts. Added a note about configuration fails if the Trion FPGA detected CRC error in volatile CRAM bit on of the new image. Corrected CBSEL description as a selection instead of address. Removed the note about some Trion FPGA have CRC check at the end of configuration. Removed the note about referring to Efinity Software User Guide and added steps on how to combine bitstream files. Added steps to create multiple image files.
May 2020	1.2	Updated device support.
May 2020	1.1	Added a topic on dual-purpose configuration pin transition time requirement.
February 2019	1.0	Initial release.