



JTAG to SPI Flash Bridge Core User Guide

UG-CORE-JTAG-SPI-FLASH-BRIDGE-v1.0

May 2025

www.efinixinc.com



Contents

- Introduction..... 3**
- Features.....3**
- Device Support..... 3**
- Resource Utilization and Performance.....4**
- Release Notes..... 4**
- Functional Description.....5**
 - Ports..... 6
- IP Manager..... 7**
- JTAG to SPI Flash Bridge Example Design..... 8**
 - Using the JTAG to SPI Flash Bridge Core with Your Board..... 9
 - Programming the Board.....10
- JTAG to SPI Flash Bridge Testbench..... 11**
- Supported Flash Devices.....12**
- Revision History.....12**

Introduction

The JTAG to SPI Flash Bridge core is an enhanced version of the JTAG SPI Flash Loader core, designed to provide seamless access to the SPI flash memory through a JTAG header on a computer. With this core, you can use the JTAG interface to program the configuration bitstream into the SPI flash memory. Once programmed, the FPGA can be reconfigured using the SPI Active mode.

Unlike its predecessor, the JTAG to SPI Flash Bridge core eliminates the need for an additional clock. It relies solely on the TCK clock, which generates all the necessary signals. This reduces the clock buffer usage, simplifies design complexity, and makes it easier to meet timing constraints. Additionally, it features an on-chip CRC32 calculation, enabling verification directly within the FPGA instead of relying on the host PC for read comparisons.

Use the IP Manager to select IP and generate files. The wizard has options to create a testbench and/or example design targeting an Efinix® development board.

Features

The JTAG to SPI Flash Bridge core includes the following features:

- Verilog RTL and simulation testbench
- Includes example designs targeting the:
 - Trion® T120 BGA324 Development Board
 - Titanium Ti60 F225 Development Board
 - Titanium Ti180 J484 Development Board
 - Titanium Ti375 C529 Development Board

Device Support

Table 1: JTAG to SPI Flash Bridge Core Device Support

FPGA Family	Supported Device
Trion	All (excluding BGA49 packages)
Titanium	All

Resource Utilization and Performance



Note: The resources and performance values provided are based on some of the supported FPGAs. These values are just guidance and can change depending on the device resource utilization, design congestion, and user design.

Table 2: Titanium Resource Utilization and Performance

FPGA	Logic Elements (Logic, Adders, Flipflops, etc.)	Memory Block	DSP Block	f _{MAX} (MHz) ⁽¹⁾	Efinity® Version ⁽²⁾
Ti60 F225 C4	1,145/60,800 (1.9%)	0/256 (0%)	0/160 (0%)	168	2025.1
Ti180 J484 C4	1,145/172,800 (0.7%)	0/1,280 (0%)	0/640 (0%)	139	2025.1
Ti375 C529 C4	1,145/362,880 (0.3%)	0/2,688 (0%)	0/1,344 (0%)	135	2025.1
Ti375 N1156 C4	1,145/362,880 (0.3%)	0/2,688 (0%)	0/1,344 (0%)	135	2025.1

Table 3: Trion Resource Utilization and Performance

FPGA	Logic Elements (Logic, Adders, Flipflops, etc.)	Memory Block	Multiplier Block	f _{MAX} (MHz) ⁽¹⁾	Efinity® Version ⁽²⁾
T120 F324 C4	1,048/112,128 (0.9%)	0/1,056 (0%)	0/320 (0%)	73	2025.1

Release Notes

You can refer to the IP Core Release Notes for more information about the IP core changes. The IP Core Release Notes are available on the [Efinity Downloads](#) page under each Efinity software release version.



Note: You must be logged in to the Support Center to view the IP Core Release Notes.

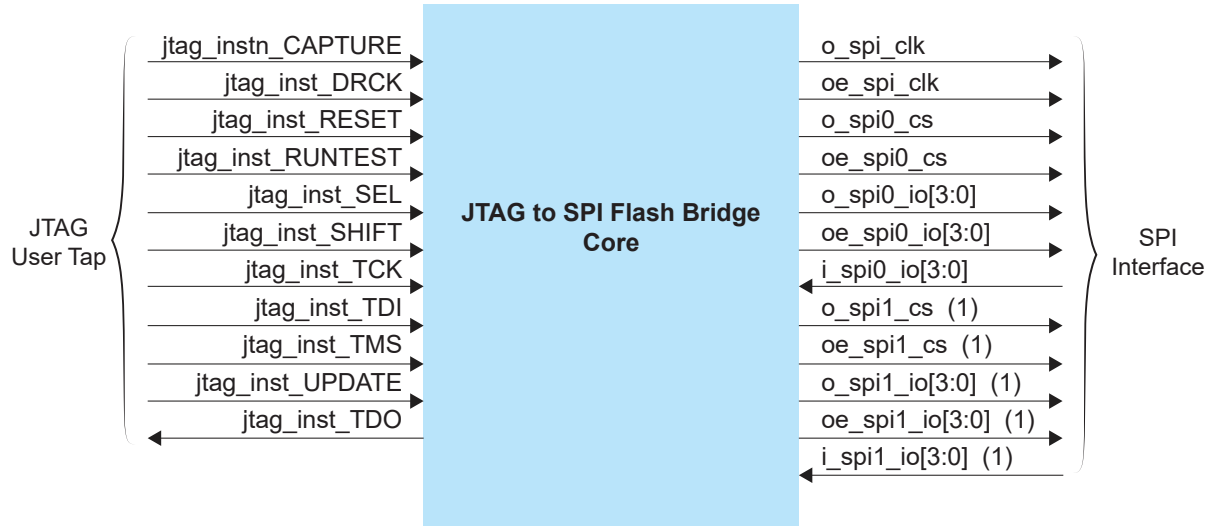
⁽¹⁾ Using default parameter settings.

⁽²⁾ Using Verilog HDL.

Functional Description

The JTAG to SPI Flash Bridge core has a JTAG user tap block interface and SPI signals on dedicated SPI interface pins.

Figure 1: JTAG to SPI Flash Bridge Core Block Diagram



Notes: (1) Applicable in dual flash mode.

Ports

Table 4: JTAG to SPI Flash Bridge Core Ports

Port	Direction	Description
o_spi_clk	Output	Clock output from FPGA CCK pin to SPI flash memory.
oe_spi_clk	Output	Output enable for o_spi_clk.
o_spi0_cs	Output	Chip select output from FPGA SSL_N pin to SPI flash memory.
oe_spi0_cs	Output	Output enable for o_spi0_cs.
o_spi0_io[3:0]	Output	Data output pins of the four bidirectional data signals, from FPGA CDI0, CDI1, CDI2, CDI3 to SPI Flash IO0, IO1, IO2, IO3..
oe_spi0_io[3:0]	Output	Output enable for o_spi0_io[3:0].
i_spi0_io[3:0]	Input	Data input pins of the four bidirectional data signals, from SPI Flash IO0, IO1, IO2, IO3 to FPGA CDI0, CDI1, CDI2, CDI3.
o_spi1_cs	Output	Chip select output from FPGA SSU_N pin to second SPI flash memory in dual flash mode ⁽³⁾
oe_spi1_cs	Output	Output enable for o_spi1_cs ⁽³⁾ .
o_spi1_io[3:0]	Output	Data output pins of the four bidirectional data signals, from FPGA CDI4, CDI5, CDI6, CDI7 to SPI Flash IO0, IO1, IO2, IO3 in dual flash mode ⁽³⁾ .
oe_spi1_io[3:0]	Output	Output enable for o_spi1_io[3:0] ⁽³⁾ .
i_spi1_io[3:0]	Input	Data input pins of the four bidirectional data signals, from SPI Flash IO0, IO1, IO2, IO3 to FPGA CDI4, CDI5, CDI6, CDI7 in dual flash mode ⁽³⁾ .

⁽³⁾ The ports are removed if targeting Trion devices.

IP Manager

The Efinity® IP Manager is an interactive wizard that helps you customize and generate Efinity® IP cores. The IP Manager performs validation checks on the parameters you set to ensure that your selections are valid. When you generate the IP core, you can optionally generate an example design targeting an Efinity development board and/or a testbench. This wizard is helpful in situations in which you use several IP cores, multiple instances of an IP core with different parameters, or the same IP core for different projects.



Note: Not all Efinity IP cores include an example design or a testbench.

Generating the JTAG to SPI Flash Bridge Core with the IP Manager

The following steps explain how to customize an IP core with the IP Configuration wizard.

1. Open the IP Catalog.
2. Choose **Memory Controllers > JTAG to SPI Flash Bridge** core and click **Next**. The **IP Configuration** wizard opens.
3. Enter the module name in the **Module Name** box.



Note: You cannot generate the core without a module name.

4. (Optional) In the **Deliverables** tab, specify whether to generate an IP core example design targeting an Efinity® development board and/or testbench. These options are turned on by default.



Note: You can migrate the provided example designs to quickly create a JTAG to SPI Flash Bridge for your own board. See [Using the JTAG to SPI Flash Bridge Core with your Board](#) for more information.

5. (Optional) In the **Summary** tab, review your selections.
6. Click **Generate** to generate the IP core and other selected deliverables.
7. In the **Review configuration generation** dialog box, click **Generate**. The Console in the **Summary** tab shows the generation status.



Note: You can disable the **Review configuration generation** dialog box by turning off the **Show Confirmation Box** option in the wizard.

8. When generation finishes, the wizard displays the **Generation Success** dialog box. Click **OK** to close the wizard.

The wizard adds the IP to your project and displays it under **IP** in the Project pane.

Generated Files

The IP Manager generates these files and directories:

- **<module name>_tpl.sv**—Verilog HDL instantiation template.
- **<module name>_tpl.vhd**—VHDL instantiation template.
- **<module name>.sv**—IP source code.
- **settings.json**—Configuration file.
- **<kit name>_devkit**—Has generated RTL, example design, and Efinity® project targeting a specific development board.
- **Testbench**—Contains generated RTL and testbench files.

JTAG to SPI Flash Bridge Example Design

You can choose to generate the example design when generating the core in the IP Manager Configuration window. Compile the example design project and download the **.hex** or **.bit** file to your board. To generate example design, the **Example Design Deliverables Option** signal must be enabled.



Important: Efinix tested the example design generated with the default parameter options only.

You can choose to generate the example design when generating the core in the IP Manager Configuration window. Compile the example design project and download the **.bit** file to your board.

The example design targets the:

- Trion® T120 BGA324 Development Board (T120 F324 development kit)
- Titanium Ti60 F225 Development Board (Ti60 F225 development kit)
- Titanium Ti180 J484 Development Board (Ti180 J484 development kit)
- Titanium Ti375 C529 Development Board (Ti375 C529 development kit)



Note: You can migrate the provided example designs to quickly create a JTAG to SPI Flash Bridge for your own board. However, some settings must be modified before you generate or compile the JTAG to SPI Flash Bridge core example design. See [Using the JTAG to SPI Flash Bridge Core with your Board](#) for more information.

Using the JTAG to SPI Flash Bridge Core with Your Board

You can use the Trion® T120 BGA324 Development Board example design as a starting point if you have a Trion FPGA on your board. Then, migrate the design as your targeted FPGA.



Learn more: Refer to the [Trion® Interactive Hardware Design Checklist and Guidelines](#), [Titanium Interactive Hardware Design Checklist and Guidelines](#), or [Topaz Interactive Hardware Design Checklist and Guidelines](#) for the SPI Active configuration pin circuitries.

Check the following settings before compiling the JTAG to SPI Flash Bridge example design.



Note: In addition to these settings, you need to verify whether Efinix supports your flash device. See [Supported Flash Devices](#) on page 12.

Dual Flash Support

If you have dual flash on your board, you need to select the **SPI Active x8 using JTAG Bridge** option when programming your board using the Efinity Programmer.

SPI Flash Resource Assignments

You need to verify the SPI flash resource assignments as follows:

Table 5: SPI Flash Resource Assignments

SPI Flash Pin	Trion Resource	Titanium/Topaz Resource
Flash#1 and #2 CLK	spi_clk	spi_clk
Flash#1 CS	spi0_cs	spi0_cs
Flash#1 IO0 (MOSI)	spi0_io[0]	spi0_io[0]
Flash#1 IO1 (MISO)	spi0_io[1]	spi0_io[1]
Flash#1 IO2 (WP#)	spi0_io[2] ⁽⁴⁾	spi0_io[2] ⁽⁴⁾
Flash#1 IO3 (HOLD#/RESET#)	spi0_io[3] ⁽⁴⁾	spi0_io[3] ⁽⁴⁾
Flash#2 CS	-	spi1_cs ⁽⁵⁾
Flash#2 IO0 (MOSI)	-	spi1_io[0] ⁽⁵⁾
Flash#2 IO1 (MISO)	-	spi1_io[1] ⁽⁵⁾
Flash#2 IO2 (WP#)	-	spi1_io[2] ^{(4) (5)}
Flash#2 IO3 (HOLD#/RESET#)	-	spi1_io[3] ^{(4) (5)}



Learn more: Refer to the device Pinouts for the actual pin names.

⁽⁴⁾ Leave unconnected if there is no connection between the FPGA and SPI flash.

⁽⁵⁾ Only required when using the SPI Active x8 in JTAG Bridge programming mode.

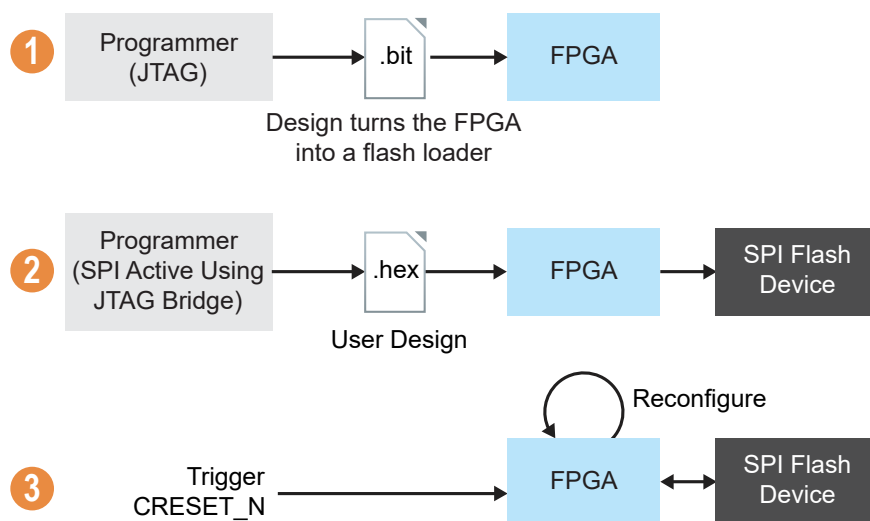
Programming the Board

Before programming your board with the example design, you need to:

- Open the generated example design project file and compile the project to generate the flash loader file **jtag2spi_flash.bit**.
- Prepare a configuration bitstream, which is also referred to as user design bitstream file, **.hex** to be programmed to the board.

The JTAG to SPI Flash Bridge core on the FPGA writes data to the flash serially (x1). You can also reconfigure the FPGA using x1, x2, x4, or x8. You can configure the settings through **Project Editor > Bitstream Generation > Program Mode**. Refer to [AN006: Configuring Trion FPGAs](#), [AN 033: Configuring Titanium FPGAs](#), or [AN 061: Configuring Topaz FPGAs](#) for more information.

Figure 2: SPI Active Using JTAG Bridge Programming Flow



The flash programming flow consists of these steps:

1. To configure the FPGA into a flash loader:
 - a. Go to Efinity Programmer, in the **Programming Mode**, select **JTAG**.
 - b. In the **Image** box, click the **Select Image File** button and select **jtag2spi_flash.bit**.
 - c. Click the **Start Program** button.
2. To program the SPI flash device through the flash loader:
 - a. Go to Efinity Programmer, in the **Programming Mode**, select **SPI Active using JTAG Bridge**.
 - b. In the **Image** box, click the **Select Image File** button and select your design file, **.hex**.
 - c. Turn on **Auto configure JTAG Bridge Image** option.
 - d. Ensure that the **Starting Flash Address** is set to **0x000000**.
 - e. Click the **Start Program** button.
3. Toggle the FPGA's **CRESSET_N** signal to trigger reconfiguration using the new flash image.



Note: You can combine step 1 and step 2 using the **SPI Active using JTAG Bridge** by enabling the **Auto configure JTAG Bridge Image** option.

JTAG to SPI Flash Bridge Testbench

You can choose to generate the testbench when generating the core in the IP Manager Configuration window. To generate testbench, the **Testbench Deliverables Option** signals must be enabled.



Note: You must include all `.v` files generated in the `/Testbench` directory in your simulation.



Important: Efinix tested the testbench generated with the default parameter options only.

Efinix provides a simulation script for you to run the testbench quickly using the Modelsim software. To run the Modelsim testbench script, run `vsim -do modelsim.do` in a terminal application. You must have Modelsim installed on your computer to use this script.

There is a Python script, `generate_tb_config.py`, which generates testbench configuration and initialization file for the SPI flash. You need to run it by entering the command `python3 generate_tb_config.py` before starting the simulation.

The testbench performs several iterations of write and read tests. The written data to the flash loader is then compared with the data read from the flash memory. Additionally, it indicates an overall pass or fail for the entire test.

After running the simulation, the test prints the following message:

```
# SUCCESS: on-chip CRC test passed
# INFO: chip reset
# JEDEC ID manufacturer ID: ef
# JEDEC ID memory type: 70
# JEDEC ID capacity: 19
# SUCCESS: SPI Flash Read/Write test passed
# SUCCESS: flash_0 test passed
# INFO: chip reset
# JEDEC ID manufacturer ID: ef
# JEDEC ID memory type: 70
# JEDEC ID capacity: 19
# SUCCESS: SPI Flash Read/Write test passed
# SUCCESS: flash_1 test passed
# SUCCESS: all tests passed
```

Supported Flash Devices

Table 6: Supported Flash Devices

Manufacturer	Family Part Number
GigaDevice	GD25Q, GD25WQ, and GD25LQ
Macronix	MX25L, MX25U, MX25V, MX75L, and MX75U
Puya Semiconductor	P25Q
Winbond	W25Q
Micron	M25P and MT25Q
XTX	XT25F
Atmel (Adesto Technologies)	AT25SF
ISSI	IS25LP128 and IS25WP512M



Note: Efinix recommends using SPI NOR flash memories.

Revision History

Table 7: Revision History

Date	Document Version	IP Version	Description
May 2025	1.0	3.0	Initial release. (DOC-1957)