



# JTAG Core User Guide

---

UG-CORE-JTAG-v1.0  
August 2025  
[www.efinixinc.com](http://www.efinixinc.com)



# Contents

- Introduction..... 3**
- Features.....3**
- Device Support.....4**
- Release Notes.....4**
- Functional Description.....5**
  - Ports.....5
- IP Manager..... 6**
- Customizing the JTAG..... 7**
- Creating a Flash Loader with the JTAG Core..... 8**
- Programming the Flash Using the Flash Loader with the JTAG Core..... 10**
- Supported Flash Devices.....11**
- Revision History.....11**

# Introduction

The JTAG core is a configurable and synthesizable Verilog module that implements the Test Access Port (TAP) state machine based on the IEEE Std. 1149.1. This IP core is intended to work in conjunction with the JTAG to SPI Flash Bridge core for flash programming, and with Efinity's Debugger for debugging purposes, particularly in scenarios where the FPGA's hard JTAG controller is disabled for security reasons.



**Note:** This core is a soft JTAG implementation and is not a hard JTAG controller in the FPGA. Thus, it only supports the following instructions:

- IDCODE
- JTAG\_USER1
- JTAG\_USER2
- JTAG\_USER3
- JTAG\_USER4
- BYPASS

Use the IP Manager to select an IP, customize and generate files. The JTAG core has an interactive wizard to help you set parameters.

## Features

The JTAG core includes the following features:

- Supports IEEE Std. 1149.1 for TAP state machine (FSM) behavior
- Configurable IDCODE instruction and value
- Supports up to 4 JTAG User TAPs
- Customizable BYPASS instruction

# Device Support

Table 1: JTAG Core Device Support

FPGA Family	Supported Device
Titanium	All
Topaz	All

## 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.

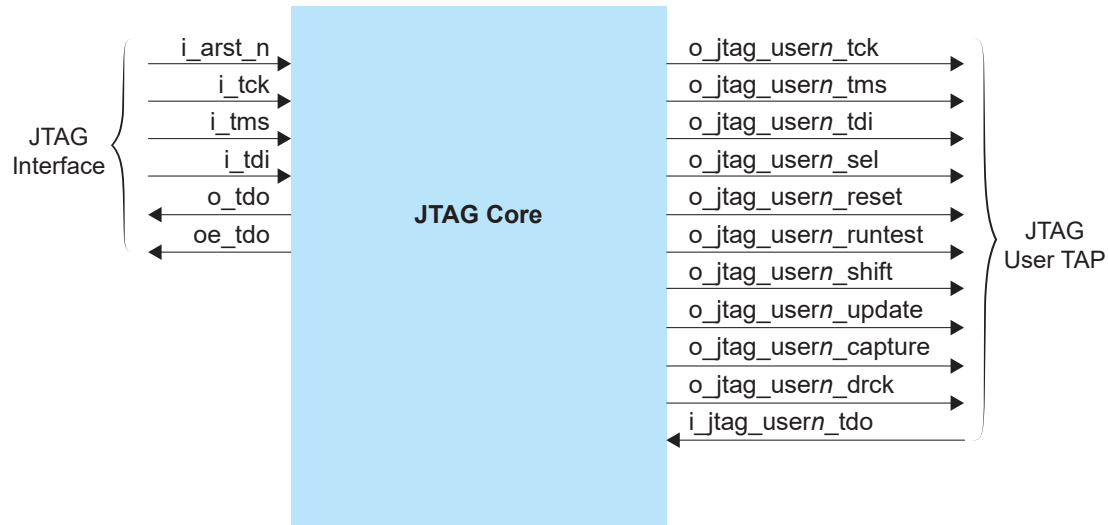


**Note:** You must be logged in to the Support Center to view the IP Core Release Notes.

# Functional Description

The JTAG core has a JTAG interface and JTAG User TAP.

Figure 1: JTAG Core Block Diagram



**Note:**  $n$  is the number for the JTAG User TAP, where  $n = 1, 2, 3,$  or  $4$

## Ports

Table 2: JTAG Core Ports

Port	Direction	Description
<code>i_arst_n</code>	Input	Asynchronous active low reset.
<code>i_tck</code>	Input	System clock.
<code>i_tdi</code>	Input	Data and instruction input pin.
<code>i_tms</code>	Input	Controls the state transitions in the TAP state machine.
<code>o_tdo</code>	Output	Data output pin.
<code>oe_tdo</code>	Output	Output enable for <code>o_tdo</code> .



**Important:** Efinix does not recommend using a regular GPIO as `i_tck` because of routing to the global clock network (GCLK), resulting in additional and variable delay.

# 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 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 **Serial Interface Protocols > JTAG** 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. Customize the IP core using the options shown in the wizard. For detailed information on the options, refer to the *Customizing the JTAG* section.
5. (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.
6. (Optional) In the **Summary** tab, review your selections.
7. Click **Generate** to generate the IP core and other selected deliverables.
8. 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.

9. 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>\_define.vh**—Contains the customized parameters.
- **<module name>\_tpl.v**—Verilog HDL instantiation template.
- **<module name>\_tpl.vhd**—VHDL instantiation template.
- **<module name>.v**—IP source code.
- **settings.json**—Configuration file.

## Customizing the JTAG

The core has parameters so you can customize its function. You set the parameters in the **General** tab of the core's IP Configuration window.

*Table 3: JTAG Core Parameters*

Parameter	Options	Description
IDCODE Instruction	-	Defines the 5-bit instruction to read the IDCODE value. Default: 5'h5
IDCODE Value	-	Defines the 32-bit IDCODE value. Default: 32'h006A0A79
Enable JTAG_USER1	Enable	Enables JTAG User TAP 1. Default: Enable
JTAG_USER1 Instruction	-	Defines the 5-bit instruction to connect to the JTAG User TAP 1. Default: 5'h8
Enable JTAG_USER2	Enable, Disable	Enable: Enables JTAG User TAP 2. Disable: Disables JTAG User TAP 2. Default: Disable
JTAG_USER2 Instruction	-	Defines the 5-bit instruction to connect to the JTAG User TAP 2. Default: 5'h9
Enable JTAG_USER3	Enable, Disable	Enable: Enables JTAG User TAP 3. Disable: Disables JTAG User TAP 3. Default: Disable
JTAG_USER3 Instruction	-	Defines the 5-bit instruction to connect to the JTAG User TAP 3. Default: 5'hA
Enable JTAG_USER4	Enable, Disable	Enable: Enables JTAG User TAP 4. Disable: Disables JTAG User TAP 4. Default: Disable
JTAG_USER4 Instruction	-	Defines the 5-bit instruction to connect to the JTAG User TAP 4. Default: 5'hB
BYPASS Instruction	-	Defines the 5-bit instructions to enable BYPASS. Default: 5'h1F

# Creating a Flash Loader with the JTAG Core

Follow these steps to create your own flash loader using the JTAG core:

1. Open the Efinity software.
2. Create a new project.
3. Assign the pins.
  - a. Open the Interface Designer.
  - b. Assign the GPIO as shown in the following tables:

**Table 4: JTAG Interface's GPIO Assignments**

FTDI Module Pin	JTAG Interface's GPIO
BDBUS0	Input: i_tck
BDBUS1	Input: i_tdi
BDBUS2	Output: o_tdo Output Enable: oe_tdo
BDBUS3	Input: i_tms



**Note:** If your JTAG interface's GPIO operates at a voltage other than 3.3 V, you need to use a voltage level translator between the FT232H/FT4232H mini module and the JTAG interface to ensure proper voltage compatibility.

**Table 5: SPI Interface's GPIO Resource Assignments**

SPI Flash Pin	SPI Interface's GPIO
Flash#1 and #2 CLK	Output: o_spi_clk Output Enable: oe_spi_clk
Flash#1 CS	Output: o_spi0_cs Output Enable: oe_spi0_cs
Flash#1 IO0 (MOSI)	Input: i_spi0_io[0] Output: o_spi0_io[0] Output Enable: oe_spi0_io[0]
Flash#1 IO1 (MISO)	Input: i_spi0_io[1] Output: o_spi0_io[1] Output Enable: oe_spi0_io[1]
Flash#1 IO2 (WP#) <sup>(1)</sup>	Input: i_spi0_io[2] Output: o_spi0_io[2] Output Enable: oe_spi0_io[2]
Flash#1 IO3 (HOLD#/RESET#) <sup>(1)</sup>	Input: i_spi0_io[3] Output: o_spi0_io[3] Output Enable: oe_spi0_io[3]

<sup>(1)</sup> Leave unconnected if there is no connection between the FPGA and SPI flash.

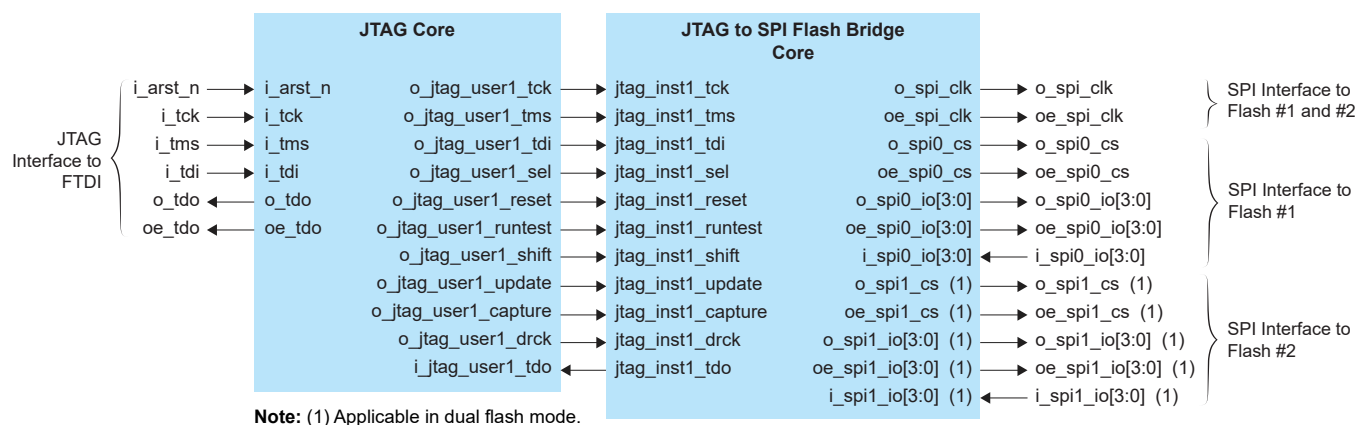
SPI Flash Pin	SPI Interface's GPIO
Flash#2 CS <sup>(2)</sup>	Output: o_spi1_cs Output Enable: oe_spi1_cs
Flash#2 IO0 (MOSI) <sup>(2)</sup>	Input: i_spi1_io[0] Output: o_spi1_io[0] Output Enable: oe_spi1_io[0]
Flash#2 IO1 (MISO) <sup>(2)</sup>	Input: i_spi1_io[1] Output: o_spi1_io[1] Output Enable: oe_spi1_io[1]
Flash#2 IO2 (WP#) <sup>(2)</sup>	Input: i_spi1_io[2] Output: o_spi1_io[2] Output Enable: oe_spi1_io[2]
Flash#2 IO3 (HOLD#/RESET#) <sup>(2)</sup>	Input: i_spi1_io[3] Output: o_spi1_io[3] Output Enable: oe_spi1_io[3]

**Note:** You need to verify whether Efinix supports your flash device. See [Supported Flash Devices](#) on page 11.

c. Click **Generate Interface Output Files**.

4. Create a new Verilog HDL source file for the top-level module and copy the top-level template from the **<project\_name>\_template.v** (available in your project's outflow folder) into this file.
5. Generate the JTAG core and JTAG to SPI Flash Bridge core, and copy the instantiation templates from the **<IP\_name>\_template.v** (available in your project's outflow folder) into the top-level module. Then, connect both instantiations according to **Figure 2: JTAG Core and JTAG to SPI Flash Bridge Connection** on page 9.
6. Constraint the design accordingly using the **.sdc** file.
7. Compile the design.

Figure 2: JTAG Core and JTAG to SPI Flash Bridge Connection

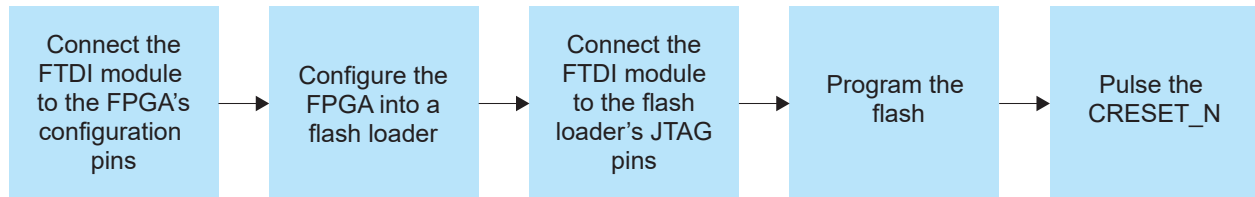


<sup>(2)</sup> Only for devices that support SPI Active x8. Check the data sheet to see if your device supports SPI Active x8.

# Programming the Flash Using the Flash Loader with the JTAG Core

The following figure shows the overall flow to program the flash on your board.

Figure 3: Flash Programming Flow



To program the flash on your board using the flash loader with the JTAG core, follow these procedures:

1. Connect the FTDI module to the FPGA's JTAG pins for JTAG configuration or the FPGA's SPI pins for SPI Passive configuration.



**Note:** In scenarios where the FPGA's hard JTAG controller is disabled, you can only use the SPI interface to configure the FPGA.

2. Configure the FPGA as a flash loader.



**Note:** See [Creating a Flash Loader with the JTAG Core](#) on page 8 to create the flash loader.

- a. Open the Efinity Programmer. In the **Programming Mode**, choose **JTAG** or **SPI Passive**, depending on your FTDI connection.
  - b. In the **Image** box, click the **Select Image File** button and select the flash loader bitstream file.
  - c. Click the **Start Program** button.
3. Connect the FTDI module to the GPIO pins assigned as the JTAG interface in the flash loader design.
  4. Program the flash.
    - a. Open the Efinity Programmer.
    - b. In the **Programming Mode**, choose **SPI Active using JTAG Bridge** for a single flash or **SPI Active x8 using JTAG Bridge** for a dual flash device.
    - c. In the **Image** box, click the **Select Image File** button and select the **.hex** bitstream file.
    - d. Turn on **Auto configure JTAG Bridge Image** and select the flash loader.



**Important:** You need to enable **Auto configure JTAG Bridge Image** option to prevent flash programming failures.

- e. Ensure that the **Starting Flash Address** is set to **0x000000**.
  - f. Click the **Start Program** button.
5. Pulse the FPGA's **CRESET\_N** signal to trigger reconfiguration using the new flash image.

# 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
August 2025	1.0	1.0	Initial release.