

# Real-Time ISP and ISP Clamp for Altera MAX Series

2014.09.22

AN-630



Subscribe



Send Feedback

This document describes the real-time in-system programmability (ISP) and ISP Clamp programming modes and their usage in the Quartus<sup>®</sup> II software, the Jam<sup>™</sup> Standard Test and Programming Language (STAPL) Player, and the Jam STAPL Byte-Code Player for Altera<sup>®</sup> MAX<sup>®</sup> II, MAX V, and MAX 10 devices.

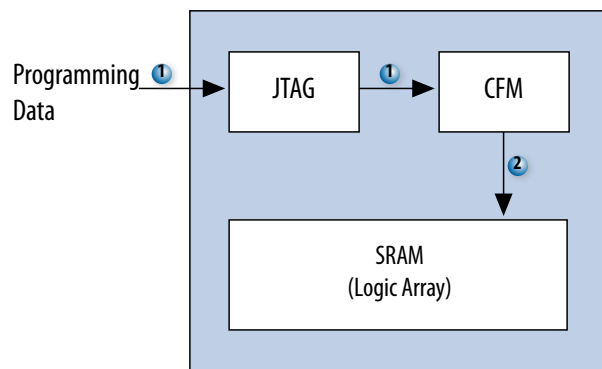
## Real-Time ISP

You can program the supported device while the device is still in operation with real-time ISP. The new design replaces the existing design only after the device goes through a power cycle. This feature enables you to perform in-field updates to the device without affecting the operation of the whole system.

## How Real-Time ISP Works

In a normal ISP operation, the new design data is downloaded from the configuration flash memory (CFM) to the SRAM after the completion of CFM programming. During the programming and downloading processes, the I/O pins remain tri-stated. After the download is completed, the device resets and enters user mode operation.

Figure 1: Normal ISP Operation



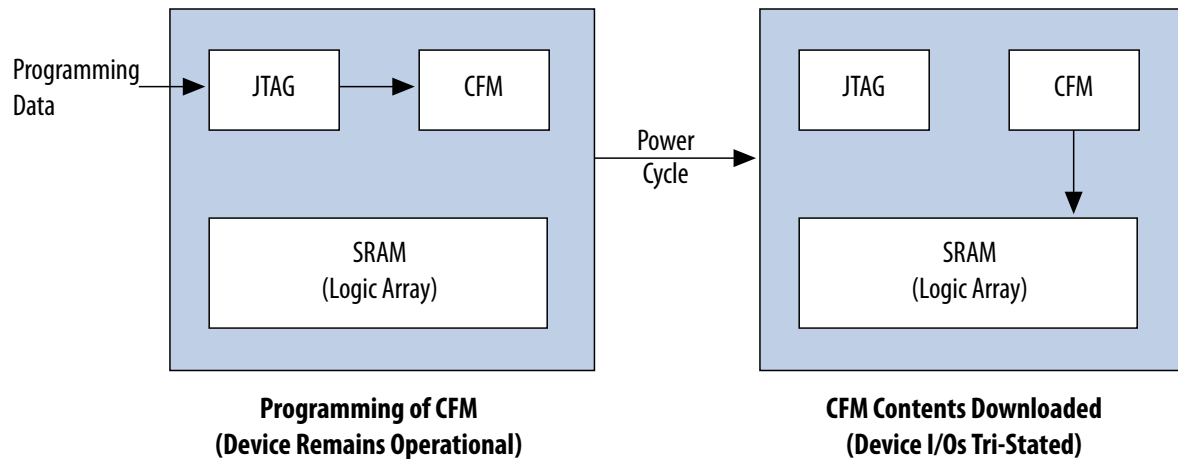
In real-time ISP mode, the user flash memory (UFM), programmable logic, and I/O pins remain operational during the progress of the CFM programming. After programming is successful, the device waits for a power cycle to occur, and then the CFM contents are downloaded to SRAM as part of a normal power-up sequence. After  $t_{\text{CONFIG}}$  time, the device enters user mode.

© 2014 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at [www.altera.com/common/legal.html](http://www.altera.com/common/legal.html). Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO  
9001:2008  
Registered

**ALTERA**<sup>®</sup>

Figure 2: Real-Time ISP Operation



#### Related Information

- [DC and Switching Characteristics for MAX II Devices](#)  
Provides more information about the tCONFIG value for a specific MAX II device.
- [DC and Switching Characteristics for MAX V Devices](#)  
Provides more information about the tCONFIG value for a specific MAX V device.
- [MAX 10 Device Datasheet](#)  
Provides more information about the tCONFIG value for a specific MAX 10 device.

## Real-Time ISP with the Quartus II Software

The Quartus II software generates these programming file formats that support the real-time ISP and ISP clamp features:

- Programmer Object File (**.pof**)
- JEDEC JESD71 STAPL Format File (**.jam**)
- JAM Byte Code File (**.jbc**)

You can use these programming files in the Quartus II Programmer. You can also use the **.jam** and **.jbc** files in other programming tools.

You must enable the real-time ISP feature before programming a device with the Quartus II Programmer. To enable this feature, in the Quartus II Programmer window, turn on **Enable real-time ISP to allow background programming**.

The device will go into real-time ISP mode when the Quartus II Programmer starts the programming operation with **.pof**, **.jam**, or **.jbc** files.

## Real-Time ISP with the Jam STAPL

You can use the **.jam** or **.jbc** created from the **.pof** to program a device in real-time ISP mode with the Jam STAPL or Jam STAPL Byte-Code Player.

- For real-time ISP with the .jam and Jam STAPL Player, type the following command at the command-line prompt:

```
jp_23 -aprogram -ddo_real_time_isp=1 <file_name>.jam
```

- For real-time ISP with the .jbc and Jam STAPL Byte-Code Player, type the following command at the command-line prompt:

```
jbi_22 -aprogram -ddo_real_time_isp=1 <file_name>.jbc
```

The names of the executable files for the players are different depending on the player version. You can download the latest version of the Jam STAPL and Jam STAPL Byte-Code Player from the Altera® web site.

## ISP Clamp

When a normal ISP operation begins on a supported device, all I/O pins are tri-stated and weakly pulled up to  $V_{CCIO}$  with internal pull-up resistors. However, there are situations when the I/O pins of the device should not be tri-stated when the device is in ISP operation. For example, in a running system, some signals such as output enable or chip enable signals might use some of the I/O pins and require those I/O pins to assume a high or low logic level, or maintain their current state when the device is in ISP mode.

In the supported devices, the ISP clamp feature allows you to use the Quartus II software to hold each I/O pin of a device to a static state when you program the device. After you successfully program the device in ISP clamp mode, the I/O pins are released and the device functions according to the new design.

You can use this feature to indicate that the device is undergoing a programming operation. When the device enters ISP clamp mode, set a particular pin to a state different than the state in the user mode operation of the device.

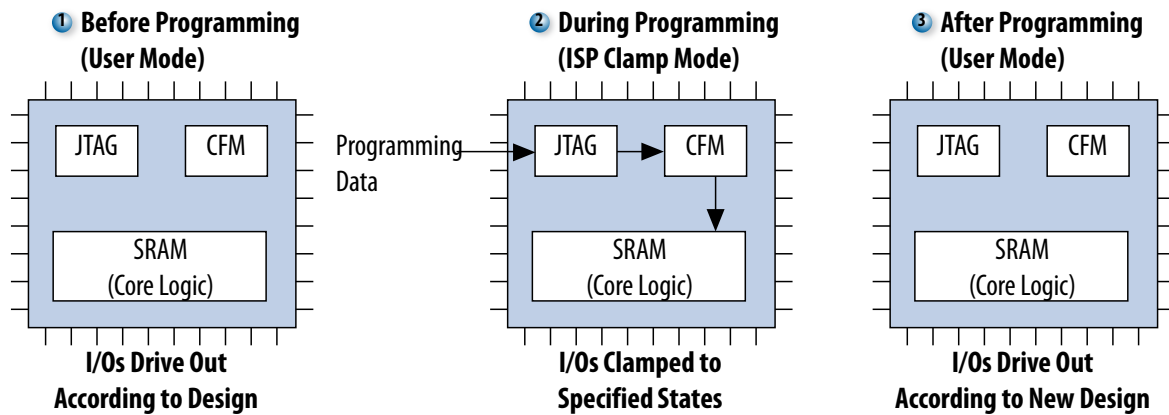
## How ISP Clamp Works

You can set the I/O pins to tri-state (default), high, or low states; or sample and sustain the existing pin state in the ISP clamp operation. The Quartus II software assigns the state values to scan into the boundary-scan registers of each I/O pin based on your settings. These values determine the state of the pins to be clamped while programming the device. When the ISP clamp feature is used, the weak I/O pull-up resistors are disabled during programming even if the I/O is clamped to a tri-state value.

Before clamping the I/O pins, the SAMPLE/PRELOAD JTAG instruction is first executed to load the values to the boundary-scan registers. The EXTEST instruction is then executed to clamp the I/O pins to the values loaded into the boundary-scan registers during SAMPLE/PRELOAD.

If you choose to sample and hold the existing state of a pin when the device enters ISP clamp mode, ensure that the signal is in a steady state. You need a steady state signal because you cannot control the sample set-up time as it depends on the  $T_{CK}$  frequency as well as the download cable and software. You might not capture the correct value if you sample a signal that toggles or is not static for long periods of time.

Figure 3: ISP Clamp Operation



## Using ISP Clamp in the Quartus II Software

To use the ISP clamp feature, you must define the states of the I/O pins. In the Quartus II software, use an I/O Pin State File (**.ips**) or the Assignment Editor to set the clamp states of the pins.

**Note:** Turn off Enable real-time ISP to allow background programming before you use the ISP clamp feature.

### Using the .ips

You can specify the clamp states of the pins with an **.ips** when the device is in ISP clamp operation without configuring the settings in the Assignment Editor and recompiling the design. The **.ips** defines the states for all the pins of the device in ISP clamp operation.

Create a new **.ips** and define the states of the pins or use an existing **.ips**. You can use the file you create to program the device with any design that targets the same device and package. Use an **.ips** together with a **.pof** that contains the programming data to program the device.

### Creating an .ips

To create an **.ips**, follow these steps:

1. On the Tools menu, click **Programmer** to open the Quartus II Programmer window. Alternatively, you can click **Programmer** on the toolbar.
2. In the Quartus II Programmer window, click **Add File** to add the programming file (**.pof**, **.jam**, or **.jbc**) into the programmer window.
3. Select the programming file in the list. Then, on the Edit menu, click **ISP CLAMP State Editor**.
4. Specify the states of the pins in your design using the ISP Clamp State Editor. There are four clamp states available—tri-state, high, low, and sample and sustain. By default, all pins are set to tri-state.
5. Save the **.ips** after making the modifications.

**Note:** Alternatively, to open the ISP Clamp State Editor to create a new **.ips**, on the File menu, point to Create/Update and click Create/Update IPS File.

## Using the .ips

To specify the **.ips** you want to use in the Quartus II Programmer, follow these steps:

1. Right-click the programming file row and select **Add IPS File**. Alternatively, click on the programming file row, and then, on the Edit menu, click **Add IPS File** to display the **Select I/O Pin State File** dialog box.
2. Select the **.ips** for your project and click **Open**.
3. The **.ips** you selected will be listed in the Quartus II Programmer window.

**Note:** Enable **ISP CLAMP** before you start programming your device.

## Saving the IPS File Information to the Programming File

You can save the pin state information in the **.ips** into the **.pof** to avoid requiring two files. You will need only the programming file to program a device in ISP clamp mode. You can also use this programming file to create the **.jam** and **.jbc** files that contain the pin state information for the ISP clamp.

To save the pin state information from the **.ips** to the programming files, follow these steps:

1. In the Quartus II Programmer, add the programming file and the **.ips**.
2. Click **Save File** and the **Save Data To File As** dialog box will appear.
3. Specify the file name and turn on **Include IPS file information**.
4. Click **Save**.

**Note:**

- The **.pof** with saved I/O pin state information supports ISP clamp operation only in the Quartus II software. For third-party tools, use **.jam** or **.jbc** files if ISP clamp is required.
- When programming a device with **ISP CLAMP** enabled, the Quartus II Programmer will first look for the **.ips**. The software will only look for the pin state information in the **.pof** if the **.ips** is not found.

## Defining the Pin States in Assignment Editor

Alternatively, you can define the pin states with the Assignment Editor and compile the design. The generated programming file will have all the pin state information.

To define the pin states with the Assignment Editor, follow these steps:

1. Click **Start Analysis and Synthesis** on the toolbar.
2. On the Assignments menu, click **Assignment Editor**.
3. In the **Category** list, select **I/O Features**.
4. In the **To** column, specify the pins that you want to clamp when the device is in ISP clamp mode. Use the Node Finder to help you select the pins.
5. Select **In-System Programming Clamp State** for all the pins in the **Assignment Name** column after you have specified the pins that you want to set state values.
6. In the **Value** column, specify the state for each pin. You can select to clamp the pins to high, low, tri-state, or to sample and sustain the pin state. By default, the pins are tri-stated when the device enters ISP clamp mode.
7. Save the assignments and recompile your design.

**Note:** After you recompile the design, the ISP clamp state information is stored in the **.pof**. You can also view the settings in the Quartus II Settings File (**.qsf**).

### Running ISP Clamp in the Quartus II Programmer

In the Quartus II Programmer, turn on **ISP CLAMP** before you program the device. Do not add any **.ips** in the Quartus II Programmer because the Quartus II Programmer will use the values in the **.ips** instead of the values in the **.pof** that you set in the Assignment Editor. The **.jam** and **.jbc** files you create using the **.pof** will have the pin state information in them.

### ISP Clamp with **.jam** or **.jbc** Files

The **.jam** or **.jbc** files for the ISP clamp contain all the pin state information and do not need any **.ips** files. Always use the **.pof** with pin state information to create the **.jam** or **.jbc** files. You can store the pin state information in the **.pof** through the Assignment Editor or by saving the pin state information to the **.pof**. You can use the **.jam** or **.jbc** files with the respective Jam STAPL or Jam STAPL Byte-Code Player, or with the Quartus II Programmer.

## Document Revision History

Date	Version	Changes
September 2014	2014.09.22	<ul style="list-style-type: none"><li>Added MAX 10 devices.</li><li>Updated template.</li><li>Restructured document.</li></ul>
December 2010	1.0	Initial release.